Research Article

Robbie T. Nakatsu*

Validation of machine learning ridge regression models using Monte Carlo, bootstrap, and variations in cross-validation

https://doi.org/10.1515/jisys-2022-0224
received August 16, 2022; accepted February 25, 2023

Abstract: In recent years, there have been several calls by practitioners of machine learning to provide more guidelines on how to use its methods and techniques. For example, the current literature on resampling methods is confusing and sometimes contradictory; worse, there are sometimes no practical guidelines offered at all. To address this shortcoming, a simulation study was conducted that evaluated ridge regression models fitted on five real-world datasets. The study compared the performance of four resampling methods, namely, Monte Carlo resampling, bootstrap, *k*-fold cross-validation, and repeated *k*-fold cross-validation. The goal was to find the best-fitting λ (regularization) parameter that would minimize mean squared error, by using nine variations of these resampling methods. For each of the nine resampling variations, 1,000 runs were performed to see how often a good fit, average fit, and poor fit λ value would be chosen. The resampling method that chose good fit values the greatest number of times was deemed the best method. Based on the results of the investigation, three general recommendations are made: (1) repeated *k*-fold cross-validation is the best method to select as a general-purpose resampling method; (2) $k = 10$ folds is a good choice in *k*-fold cross-validation; (3) Monte Carlo and bootstrap are underperformers, so they are not recommended as general-purpose resampling methods. At the same time, no resampling method was found to be uniformly better than the others.

Keywords: ridge regression, machine learning, model validation, cross validation, resampling methods

1 Introduction

In machine learning, linear regression is one of the most widely used techniques for building a model that predicts, or estimates, a quantitative outcome. Numerous textbooks and articles have been written on the subject. At its core, linear regression involves fitting a linear model that minimizes the sum of squared error (SSE), and then uses the linear model to make predictions on unseen data. Given a linear regression model of the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon, \tag{1}$$

where Y is the target or outcome variable, and X_1, X_2,..., X_p are the independent, or feature variables, ordinary least squares (OLS) regression will find the β coefficients that minimize the SSE.

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

where y_i is the actual outcome and $\hat{y}_i$ is the predicted outcome fitted by using the regression model for $i = 1$ to n observations.

* **Corresponding author: Robbie T. Nakatsu,** Department of Information Systems and Business Analytics, Loyola Marymount University, Los Angeles, CA 90045, USA, e-mail: Robbie.Nakatsu@lmu.edu
ORCID: Robbie T. Nakatsu 0000-0003-1523-4224

However, linear regression can be prone to model overfitting. This can especially be problematic when fitting a complex model having many parameters. Model overfitting means that a regression model does not generalize well beyond the dataset on which it was trained, or as Provost and Fawcett [1, p. 113] define it, "a tendency of data mining procedures to tailor models to the training data, at the expense of generalization to previously unseen data points." A visualization of model overfit involving linear regression is provided by Figure 1. A set of x values ($n = 50$) is randomly generated from a normal distribution ($\mu = 50$, $\sigma = 10$). The y values are calculated from the quadratic function: $\boldsymbol{y = x^2 - 5x + \epsilon}$, with the error term ϵ also randomly generated from a normal distribution ($\mu = 0$, $\sigma = 10$). The data are then fit with four different regression models: (1) a straight line (underfit model), (2) a quadratic curve (correct fit), (3) a polynomial of degree 10 (overfit model), and (4) a polynomial of degree 15 (overfit model). Figure 1 shows how models (3) and (4) overfit the data. Both models chase after noise in the data, and result in more erratic curves. When a polynomial of degree 15 is fit to the data, the curve becomes extremely erratic. Even though R^2 continues to improve with higher order polynomials, the overfit models would not generalize well to unseen data.

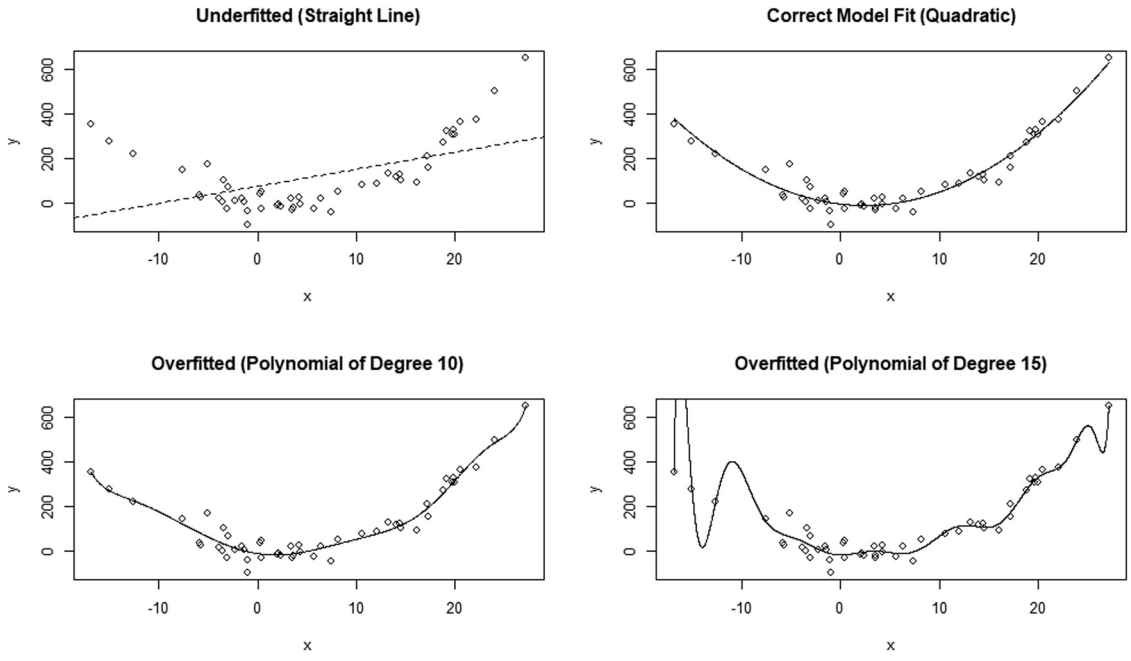

Figure 1: Four regression models are fitted on the same set of data. On the upper left panel, a line underfits the data ($R^2 = 0.269$). On the upper right panel, a correct quadratic model is fitted ($R^2 = 0.938$). On the bottom left, a polynomial of degree 10 is fitted ($R^2 = 0.945$) and on the bottom right, a polynomial of degree 15 is fitted ($R^2 = 0.954$). Even though the R^2 continues to improve, the higher degree polynomials are overfit. Source: Nakatsu [4].

One technique that can be used to deal with model overfitting is known as ridge regression. Hoerl and Kennard [2], Marquardt [3], and others originally proposed the technique as a way for a regression model to achieve better predictive accuracy in the presence of multicollinearity. The technique involves fitting a model of all predictors, like in OLS regression, but the estimated coefficients are shrunken towards zero. To accomplish this, ridge regression penalizes the β parameter estimates by adding a penalty term to the SSE in equation (2).

$$\text{SSE} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p}\beta_j^2, \tag{3}$$

where i is the observation from 1 to n and β_j is the coefficient of the predictor variable from $j = 1$ to p.

The result is that ridge regression shrinks the β estimates toward 0 as λ becomes large. Conversely, the β estimates of ridge regression are the same as the β estimates of OLS regression when $\lambda = 0$. The effect of

shrinking the coefficients (a process known as regularization) is that the regression model is less prone to model overfitting. Selecting and fine-tuning the λ parameter can be accomplished by finding the ridge regression model that minimizes prediction error – in regression modeling, mean squared error (MSE) is commonly used. MSE is the average of the squared residuals or equation (2) divided by n.

A typical method is to randomly split a dataset into two sets, a training set and a validation set. A ridge regression model is fit on the training set, and then validated on the validation set. The validation process calculates the MSE over a range of λ values. The λ value that results in the lowest MSE on the validation set is chosen as the best-fitting λ. Because the validation dataset is "held-out," this method is sometimes referred to as the holdout method.

Unfortunately, this holdout technique provides only a single estimate of a model's validation error. The split between training and validation sets could be a particularly biased choice – even if randomized – and could either underestimate or overestimate MSE. This could especially pose a problem when dealing with smaller datasets. A way to circumvent this problem is to use **resampling**, which means repeatedly drawing randomized samples from a dataset and refitting the model on each sample [5]. By resampling, the average error rate of multiple runs can provide a better estimate of error rate than a single-point estimate. This is the approach that is investigated in this study. To that end, four of the most common resampling methods are compared: (1) Monte Carlo resampling, (2) bootstrap, (3) k-fold cross-validation (k-fold CV), and (4) repeated k-fold CV.

Among resampling methods, the most straightforward is **Monte Carlo** resampling. This method is also known as repeated learning-testing [6], repeated holdout [7], or random subsampling [8]. The method involves randomly generating training/validation splits on the same dataset multiple times. In this study, a 75%/25% (training/validation) split is used. Monte Carlo resampling can be implemented first by shuffling the dataset (i.e., randomizing the rows) and then designating the first 75% of the rows as the training set, and the remaining 25% as the held-out validation set. For each repetition, a ridge regression model is fit on the training set and then validated on the validation set. The average validation error rate is then calculated over the repetitions.

Bootstrap also randomly generates training and validation sets but, unlike Monte Carlo resampling, the rows of the training set are randomly selected from the dataset with replacement [9]. Because sampling is performed with replacement, a bootstrap sample will contain rows that are duplicated – on average 63.2% of the rows are selected; by the same token, there will be some rows that are not chosen – on average the remaining 36.8% rows are not selected. The chosen rows are designated as the training set, while the unchosen rows become the validation set. Like Monte Carlo resampling, bootstrap is intended to be run multiple times so that the average validation error across the repetitions can be used to estimate error rates more accurately.

***k*-fold CV** is a method that was introduced in 1974 [10–12] and over the years has become a popular resampling method. Its popularity is evidenced in the many practitioner guides and textbooks, published within the last 10 years, that advocate its use in model validation and selection [1,5,13–19]. The method begins by randomly splitting a dataset into k partitions called folds (5 or 10 folds are most commonly used). Subsequently, the technique iterates k times: for each iteration, one fold is set aside as the validation set, and the remaining k–1 folds are used to train the model. The model thus built is validated on the validation set. After iterating k times this way, an average of the k validation errors is calculated so that a more accurate estimate of error rate can be obtained.

The most extreme case of k-fold CV is known as **leave-one-out cross-validation** (LOOCV). In this approach, a single observation is held out as the validation set and the remaining observations are used as the training set (i.e., the fold size is 1). The procedure is repeated n times, where n represents the number of samples in the dataset. The average of the n validation errors is used as the estimate for the error rate. Because LOOCV is run n times, it can become computationally prohibitive for larger datasets. In addition, prior research indicates that LOOCV results in estimates that have high variance, leading to unreliable estimates [20,21]. For these reasons, LOOCV is not further investigated in this study.

Another variation in cross-validation is known as **repeated *k*-fold CV**. Under this method, k-fold CV is repeated multiple times and the average of the multiple repetitions is used to estimate the error rate. Given n repetitions, there will be $n \times k$ validation errors; hence, the average validation error is calculated over the $n \times k$ repetitions. The most common way of running k-fold CV is only once; thus, the repeated method has been suggested by others as a way of obtaining more accurate and reliable estimates of error rates [17].

This research investigation seeks to understand the resampling methods in greater detail and offer some specific guidelines on its usage. Indeed, in recent years, there have been several calls for more practical guidelines and more transparency. In a recent paper, King et al. [22] wrote: "ML is now a key technology in modern science. However, its techniques need to be better understood. We therefore call for a dialogue between ML and domain scientists in which ML methods, such as cross-validation, can be explained to domain scientists so that they can trust and benefit from them." We could not agree more with this statement. The current literature on resampling methods such as cross-validation is confusing, and sometimes contradictory; worse, there is sometimes no practical guidance at all. Here are some current issues.

Sometimes no guidance at all is offered in the most popular practitioner guides and textbooks. While some books do an excellent job in explaining and illustrating the resampling methods themselves [5,15,16] they offer little to no guidance on how to choose among the approaches. The practitioner is left to adopt a trial-and-error approach when choosing among the methods.

Many of the practitioner guides and studies that look at resampling arrive at conflicting recommendations. For example, some recommend LOOCV in some cases, specifically when computationally feasible [17,23,24], while others suggest that LOOCV should be avoided altogether [14,20]. The fold size k is also not well understood, and the recommendations are conflicting. The most common recommendation for fold size k is 5 or 10 [8], while others claim that other values can be used [25]. In one research study that examined the selection of SVM hyperparameters, the recommendation was to use $k = 2$ [26].

Finally, in a review of the research literature, it appears that single-run k-fold CV is the most popular validation method to use when tuning hyperparameters on a machine learning method. Most research studies do not even consider repeated k-fold CV, even if it could potentially benefit its results [27–30]. On the other hand, other studies have suggested that repeated k-fold CV could be beneficial [7,31], while one study recommends against its use [32].

To address the confusion surrounding resampling methods, a simulation study will look at and evaluate these different resampling methods on five different datasets, all involving a regression task in which a quantitative outcome is to be predicted from a set of features. A ridge regression model is fitted on each dataset and the regularization parameter λ is tuned using resampling. Of particular importance is how well the four resampling methods perform in selecting a good λ value. The following questions are investigated:

(1) Which of the four resampling methods is most effective in selecting a suitable regularization parameter λ?
(2) Does increasing the number of repetitions – from 10 to 50 – improve the performance of the resampling method?
(3) Keeping the number of repetitions constant, which approach, single-run cross-validation or repeated cross-validation, performs better?
(4) For k-fold CV, what is an appropriate fold size k?
(5) Which randomization approach is more effective, Monte Carlo (sampling without replacement) or bootstrap (sampling with replacement)?

2 Related work

Over the last decades, several researchers have investigated and compared the performance of different resampling methods. Many of the earlier studies, especially, involve experimental studies on artificial and smaller datasets. Later studies look more systematically at variations in the four resampling methods and take on more computationally intensive approaches.

Some of the earlier studies report positive results on bootstrap's performance. Efron [21], for example, reports on sampling experiments comparing LOOCV to bootstrap. He found that LOOCV gives nearly unbiased estimates of error but often with high variance, particularly if the sample size is small. Moreover, he found that bootstrap performed best in his experiments. However, he recommended bootstrap with caution, pending further numerical and theoretical study. Delaney and Chatterjee [33, p. 261], likewise,

advocate the use of bootstrap and note that its benefits include its less subjective nature, ease of implementation, and robustness: "The bootstrap choice of the ridge parameter can be justified because it is based on repeated and independent estimates of multiple predictions and is, therefore, robust." However, they compare bootstrap only to the ridge trace method – a method that is not addressed as this study is focused on comparisons among more general-purpose resampling methods.

Later research investigations extend the evaluation of resampling methods to k-fold CV and Monte Carlo resampling. Using a simulation study, Burman [6] evaluates three methods: LOOCV (referred to as ordinary cross-validation), k-fold CV, and Monte Carlo (referred to as repeated learning-test). His recommendation is to use k-fold CV or Monte Carlo if the computational cost of LOOCV is large. Further, with respect to fold size, he advocates the use of larger fold sizes because the bias and the variance of k-fold CV estimate decreases as the value of k increases. Regarding Monte Carlo, he does not advocate its use over k-fold CV because k-fold CV has smaller variance.

Breiman and Spector [20] look at submodel selection in regression – their task is about feature selection in a regression model, which is a different task from regularization parameter tuning, but their study's results are illuminating, nonetheless. Their study involved an extensive simulation. They discovered that non-resampling estimates such as C_p and adjusted R^2 turn out to be highly biased methods for submodel selection. According to their results, the two best resampling methods to use in submodel selection are k-fold CV and bootstrap. One of their findings was that 5-fold and 10-fold CV is better at submodel selection and evaluation than LOOCV.

Kohavi [8] reports on a large-scale experiment using classification algorithms C4.5 (decision trees) and Naïve Bayes on real-world datasets. He used resampling to estimate the effects of different parameters on these classification algorithms. He compared k-fold CV to bootstrap. For k-fold CV, he varied the number of folds and whether the folds are stratified or not (by stratification he means that the folds contain approximately the same proportion of outcome labels as the original dataset). For bootstrap, he varied the number of samples. His main result was that 10-fold CV is better than bootstrap in model selection. In addition, his results show that stratification is generally a better scheme, both in terms of bias and variance when compared to regular (non-stratified) k-fold CV. (In this study, stratified k-fold sampling is always used, because it has become standard practice in machine learning model validation.)

More recent studies have looked at more computationally intensive methods of cross-validation, including repeated k-fold CV, and compared it to more traditional approaches. Molinaro et al. [31] studied classification problems using the algorithms linear discriminant analysis, diagonal discriminant classifiers, nearest neighbors (NN), and CART. Their study used microarray datasets (in genomic studies) in which there are thousands of features (i.e., gene measurements) collected on relatively few samples (i.e., patients). The goal of their analysis was to find differences among resampling methods in the estimation of generalization error. They examined the effect of repeated k-fold CV using 2, 5, and 10 folds. Each was repeated 10 and 30 times. One of their findings was that repeated k-fold CV is beneficial: when increasing repetitions from 1 to 10, they found significant improvement in classifier performance; however, there was minimal improvement when reps were increased from 10 to 30. A second finding was that Monte Carlo did not decrease bias to warrant its use over k-fold CV. A third finding was that as the sample size grows, the differences in performance among the resampling methods decrease.

Nakatsu [34] also performed an evaluation of resampling methods on four classification algorithms: support vector machines (SVM), random forests, k-NN, and decision trees. Variations in resampling methods were used to tune parameters on the classifiers. Nakatsu found significant differences in performance among the resampling methods. No one resampling method was found to be always better than the others, but repeated k-fold CV was, overall, the best performer across all four classification algorithms.

The current study continues in the tradition of these prior research investigations. Most of the prior studies involve classification tasks, but none report on fine-tuning the regularization parameter in a regression model. Because of the popularity of ridge regression with the machine learning community, this study explores whether prior research results would hold in the ridge regression context. Second, this study investigates any unique characteristics of ridge regression modeling in which the general results might not hold.

Finally, this research only considers standard resampling methods – i.e., Monte Carlo, bootstrap, and k-fold CV. It does not investigate how to modify these resampling approaches themselves to obtain better

ridge regression parameter estimates. Algamal [35] proposes a modified version of the cross-validation approach, one which repeats the fold assignment and then determines a "quantile value" that is considered as the final optimal value. In a second study [36], Algamal proposes a "kidney-inspired algorithm," which is a population-based algorithm inspired by the kidney process in the human body. Modifying the resampling methods, themselves, is not considered in this research study.

3 Methods and materials

The simulation study looked at five datasets of varying sizes, drawn from different application domains. All the following datasets are available online:

- Baseball [5]. Major League baseball data from 1986 to 1987 seasons. Target variable: salary
- Boston [37]. Housing values in the suburbs of Boston. Target variable: median home value
- Concrete [38]. Concrete mixtures and their compressive strengths. Target variable: compressive strength.
- Parkinson's [39]. Biomedical voice measurements from 42 people with early-stage Parkinson's disease. Target variable: Total-Unified Parkinson's Disease Rating Scale (UPDRS).
- Superconductor [40]. Superconductors, and their properties. Target variable: critical temperature.

A ridge regression model was fit, separately, on each of the five datasets. Each modeling task involved the prediction of a numeric outcome from a feature set. The datasets range in size from $n = 263$ to 21,263 rows and from $p = 9$ to 81 features. They also vary in terms of subject matter, including a dataset from business (Baseball), socioeconomics (Boston), medicine (Parkinson's), materials science (Concrete), and physical science (Superconductor). The intent was to develop generalized findings across a range of datasets in terms of size and problem-solving domain. Furthermore, the goal was to demonstrate these findings on real-world datasets, not simulated data.

3.1 Determining good fit, average fit, and poor fit λ values

To determine how well the four resampling methods performed, ridge regression was run multiple times to determine good fit, average fit, and poor fit values of λ on each of the five datasets. First, 10-fold CV was performed 50–100 times on each of the 5 datasets to obtain a rough sense of where the optimal value of λ lay, based on MSE. Then, a determination was made as to what an appropriate range of λ values would be to look at further. For example, on the Boston dataset, it was determined that the optimal λ value would be low, so λ values ranging from 0.01 to 1.0 (in 0.01 increments) were tested. For the Baseball dataset, a different set of λ values was used – from 0.2 to 20.0 (in 0.2 increments). To fine-tune the MSE associated with each of the λ values, 10-fold CV was repeated multiple times until we could discern what the good fit, average fit, and poor fit values were. Figure 2 shows the resulting data plots when 10-fold CV was repeated. Figure 2(a) shows the average MSE on the Boston dataset after 500 reps of 10-fold CV were run. You can see that there is still noise in the data after 500 reps. When the number of reps was increased to 10,000, we were confident where the best-fit values lay, as the MSE data converges to follow a curved line more closely, Figure 2(b).

Once an MSE curve was found, the following criteria were used to determine the fit type: (1) the top 20% λ values (i.e., associated with the lowest MSE values) were designated as a **good fit**; (2) the middle 60% λ values were designated **average fit**; and (3) the bottom 20% λ values (i.e., associated with the highest MSE values) were designated as **poor fit**. The good fit and poor fit intervals are indicated in Figure 2(b); all remaining λ values are designated average fit.

The same procedure and criteria were applied to the other four datasets to determine good fit, average fit, and poor fit λ values. For example, Figure 2(c) shows the MSE curve for the Baseball dataset when 75,000 reps of 10-fold CV were performed, as well as the designation of good fit and poor fit values on the graph. Good fit values lay between $\lambda = 4.6$ and $\lambda = 8.4$; poor fit values lay between $\lambda = 0.2$ and $\lambda = 2.0$ as well as between $\lambda =$

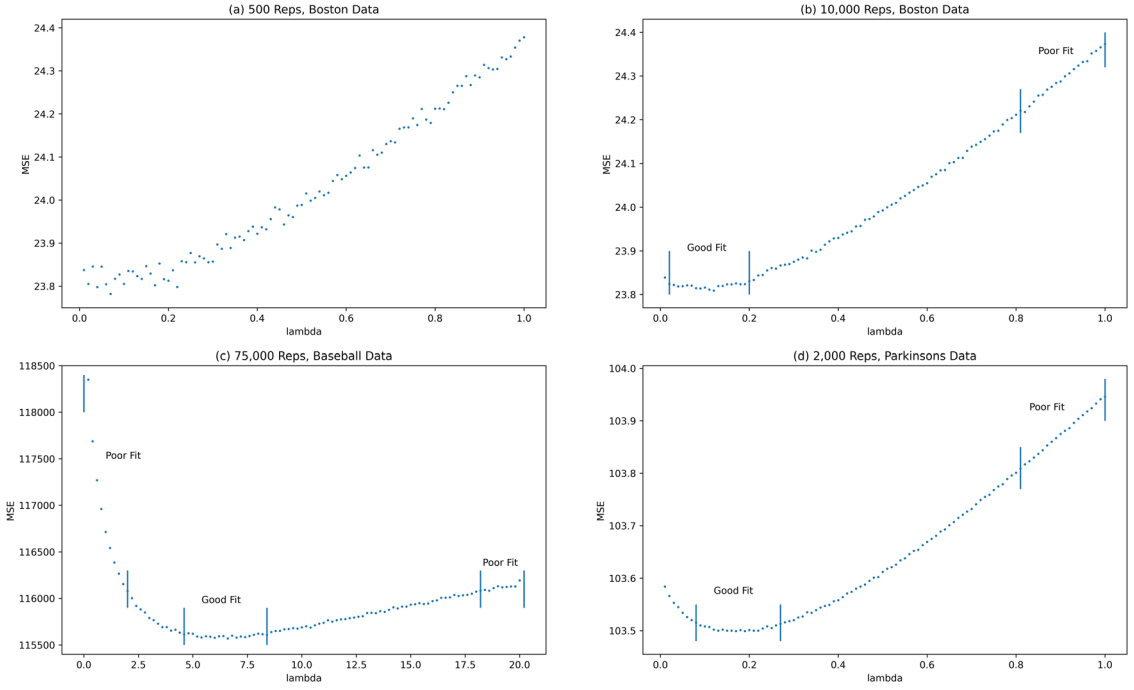

Figure 2: Average MSE over a range of lambda (λ) values. The four plots show average MSE for a range of λ values. Average MSE is calculated over multiple reps of 10-fold CV. Plot (a) shows average MSE for 500 reps on the Boston dataset. Plot (b) shows average MSE for 10,000 reps on the Boston dataset. With more reps, the data in Plot (b) has less noise and approaches a curved line. Plot (c) shows average MSE for 75,000 reps on the Baseball dataset. Plot (d) shows average MSE for 2,000 reps on the Parkinson's dataset.

18.2 and $\lambda = 20.0$; all other values were designated average fit. Finally, Figure 2(d) shows the MSE curve for the Parkinson's dataset, where, again, good fit and poor fit intervals are indicated. Note that the Parkinson's dataset required only 2,000 reps to converge – this can be seen from the smooth curve in Figure 2(d) – whereas the Boston and Baseball datasets required significantly more reps, 50,000 and 75,000, respectively.

It is important to underscore that even though repeated 10-fold CV was used to determine the good fit, average fit, and poor fit λ values, the same solution could be arrived at using multiple iterations of either Monte Carlo or bootstrap. This issue is addressed in Appendix A, which illustrates how Monte Carlo resampling would arrive at the exact same solution as that found by repeated k-fold CV. The reason repeated 10-fold CV was used instead is that it arrived at the final solution much more efficiently than either Monte Carlo or bootstrap could.

3.2 Evaluating the four resampling methods

Once fit types (good, average, and poor) were established and determined, we were ready to begin evaluation of the resampling methods. Nine variations of the four resampling methods were evaluated, by varying either the number of reps (Monte Carlo and bootstrap), the number of folds (k-fold CV), or both the number of folds and reps (repeated k-fold CV):

- Monte Carlo: (1) 10 reps; (2) 50 reps
- bootstrap: (3) 10 reps; (4) 50 reps
- k-fold CV: (5) 10 folds; (6) 50 folds
- repeated k-fold CV: (7) 5-fold CV, 2 reps; (8) 10-fold CV, 5 reps; (9) 5-fold CV, 10 reps

For repeated k-fold CV, we looked at one 10-iteration approach (5-fold CV, 2 reps) and two 50-iteration approaches (10-fold CV, 5 reps and 5-fold CV, 10 reps). We sought to make comparisons, separately, among the 10-iteration approaches and among the 50-iteration approaches. Comparing a 50-iteration approach to

a 10-iteration approach would be like comparing apples to oranges (a higher iteration approach was likely to do better simply because more repetitions mean more accurate estimates of MSE) and we wanted to control for this separately.

For each of the nine resampling variations, 1,000 runs were performed to see how often a good fit, average fit, and poor fit λ value would be chosen. For both Monte Carlo and bootstrap, either 10 or 50 random samples were drawn. For example, for the Monte Carlo 10 reps approach, a randomized training/validation split (75%/25% split) was generated 10 times and the average MSE was calculated over the 10 runs. This was done for each λ value over a designated range of values. The λ value that resulted in the lowest average MSE was selected – i.e., the chosen λ. We counted how many times the chosen λ was a good fit, average fit, or poor fit over 1,000 runs. See Algorithm 1 below for the steps involved in Monte Carlo and bootstrap.

A similar approach is used when evaluating k-fold CV and repeated k-fold CV. For example, for 5-fold CV 10 reps, the dataset was randomly split into 5 folds and the MSE was calculated 5 times. This was then repeated 10 times, resulting in 5×10 or 50 MSE calculations. The average MSE was calculated over the 50 runs. This was repeated for each of the λ values. The λ value that resulted in the lowest average MSE was selected as the chosen λ. Again, we counted how many times the chosen λ was a good fit, average fit, or poor fit over 1,000 runs. The steps for k-fold CV, and repeated k-fold CV are given in Algorithm 2 below.

ALGORITHM 1: Monte Carlo and Bootstrap

1. Repeat for λ from 0.01 to 1.0, in 0.01 increments[1]:
   A. Repeat n times:
      a) Randomly divide[2] the whole dataset into a training set and a validation set.
      b) Build a ridge regression model on the training set using the λ value.
      c) Validate the model on the validation set by calculating the MSE.
   B. Calculate the average MSE for each MSE calculated in
      Step 1.A.c over the n repetitions. This value represents the average MSE for the λ value.
2. Select the λ value that has the lowest average MSE calculated in Step 1B. Refer to this value as the **chosen** λ.

Footnotes:

[1] The λ values are different, depending on the dataset. Here the λ values for the Boston dataset are used.
[2] The way the randomization takes place in Step 1.A.a is either Monte Carlo (75%/25% training/test split) or bootstrap (sampling with replacement). For bootstrap, the validation set comprises the unselected rows.

ALGORITHM 2: k-Fold CV and Repeated k-Fold CV

1. Repeat for λ from 0.01 to 1.0, in 0.01 increments[1]:
   A. Repeat n times[2]:
      a) Randomly generate k folds from the entire dataset.
      b) Repeat for folds 1 through k:
         i) The current fold selected is the validation set; all other k–1 folds are the training set.
         ii) Build a ridge regression model on the training set using the λ value.
         iii) Validate the model on the validation set by calculating the MSE.
   B. Calculate the average MSE for each MSE calculated in
      Step 1.A.b.iii, over the k runs of k-fold CV and repeated n times. There will be a total of $n*k$ validation errors. This value represents the average MSE for the λ value.
3. Select the λ value that has the lowest average MSE calculated in Step 1.B. Refer to this value as the **chosen** λ.

Footnotes:

[1] The λ values are different, depending on the dataset. Here the λ values for the Boston dataset are used.
[2] n is 1 for single-run k-fold CV, but otherwise indicates the number of reps for repeated k-fold CV.

4 Results

The main results of the study are presented in Table 1. The table summarizes how well each of the nine resampling approaches performed over the five datasets, in terms of how often a good fit, average fit, and poor fit λ was chosen. The methods are grouped according to whether they are 10-iteration approaches or 50-iteration approaches so that baseline comparisons can be made within each of these groups.

Table 1: Fit classification results for nine resampling approaches

	10-iteration approaches				50-iteration approaches				
	(1) Monte Carlo 10 reps	(2) Bootstrap 10 reps	(3) 10-fold CV	(4) 5-fold CV 2 reps	(5) Monte Carlo 50 reps	(6) Bootstrap 50 reps	(7) 50-fold CV	(8) 5-fold CV 10 reps	(9) 10-fold CV 5 reps
Baseball									
***n* = 263**									
Good fit	208	172	300*	285*	205	174	237	330*	359*
Average fit	627	655	589	613	652	669	628	606	575
Poor fit	165	173	111	102	143	157	135	64	66
Boston									
***n* = 506**									
Good fit	218	173	635**	589	270	232	513	682	750**
Average fit	619	632	365	411	634	673	480	318	250
Poor fit	163	195	0	0	96	95	7	0	0
Concrete									
***n* = 1,030**									
Good fit	444	517	965*	954*	646	681	985	989	998**
Average fit	528	446	35	46	351	318	15	11	2
Poor fit	28	37	0	0	3	1	0	0	0
Parkinson's									
***n* = 5,875**									
Good fit	276	277	823*	796*	338	351	916*	911*	922*
Average fit	587	636	177	204	610	616	84	89	78
Poor fit	137	87	0	0	52	33	0	0	0
Superconductor									
***n* = 21,263**									
Good fit	700	772	1,000*	1,000*	941	962	1,000	N/A	N/A
Average fit	290	226	0	0	59	38	0		
Poor fit	10	2	0	0	0	0	0		

**Denotes single best performance, * denotes tied best performance.

A first glance through Table 1 reveals significant differences in performance among the resampling approaches. For example, in looking at the Parkinson's dataset (fourth row of Table 1), Monte Carlo and bootstrap methods performed poorly – for the 10-iteration approaches, they selected a good fit λ value only 27.6 and 27.7% of the time, respectively, whereas the 10-fold CV and 5-fold CV 2 reps approaches performed dramatically better, selecting a good fit λ value 82.3 and 79.6% of the time, respectively. Also, indicated in the table is the single best performer (denoted by **) or tied best performer (denoted by *). This is indicated, separately, for the 10-iteration approaches and the 50-iteration approaches. The difference between "single best" and "tied best" is that a resampling approach is deemed single best if the Chi-square statistic between the top performer and second-best performer is significant ($p < 0.05$), whereas a tied best performer means the statistic is not significant. You can quickly observe in Table 1 that the best performers are consistently k-fold CV and repeated k-fold CV for both 10-iteration and 50-iteration approaches; Monte Carlo and bootstrap are never the best performers across all five datasets.

In this section, five specific questions regarding differences in performance in the resampling approaches are explored in more detail:

(1) Did performance increase when reps (iterations) were increased from 10 to 50? (Section 4.1)
(2) What was the effect of different values of k in k-fold CV? (Section 4.2)
(3) Keeping the number of iterations constant, which method was superior, single-run CV or repeated CV? (Section 4.3)
(4) Which method of randomization was superior, Monte Carlo or bootstrap? (Section 4.4)

To compare the performance of one resampling approach to another, Pearson's Chi-square test is calculated. This statistic compares the frequencies observed in certain categories to the frequencies you might expect to get in those categories simply by chance. In this analysis, if the statistic is significant ($p < 0.05$), then there is a significant association between the resampling method and its performance (i.e., how often it chooses good fit, average fit, and poor fit). Refer Tables 2–5 in the Sections 4.1–4.4 for the Chi-square test results.

Table 2: Chi-Square tests, comparing 10 reps to 50 reps

	(1) Monte Carlo, 10 reps vs Monte Carlo, 50 reps	(2) Bootstrap, 10 reps vs bootstrap, 50 reps	(3) 10-fold CV vs 50-fold CV	(4) 10-fold CV vs 10-fold CV, 5 reps	(5) 5-fold CV, 2 reps vs 5-fold CV, 10 reps
Baseball	2.1	0.9	↓11.0**	16.9***	17.4***
Boston	23.1***	44.4***	↓30.4***	31.1***	18.8***
Concrete	93.2***	78.0***	8.2**	30 0***	22.1***
Parkinson's	44.9***	33.3***	38.1***	44.1***	52.9***
Superconductor	198.3***	156.5***	0.0	N/A	N/A

*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

Table 3: Chi-Square tests, comparing 5 folds to 10 folds

	5-fold CV, 2 reps vs 10-fold CV	5-fold CV, 10 reps vs 10-fold CV, 5 reps
Baseball	1.2	2.1
Boston	4.5*	11.0***
Concrete	1.6	5.0*
Parkinsons	2.4	0.8
Superconductor	0.0	N/A

*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

Table 4: Chi-Square tests, comparing single-run CV to repeated CV

	(1) 50-fold CV vs 5-fold CV, 10 reps	(2) 50-fold CV vs 10-fold CV, 5 reps
Baseball	41.0***	51.0***
Boston	59.4***	120.7***
Concrete	0.6	10.0**
Parkinson's	0.2	0.2
Superconductor	N/A	N/A

*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

Table 5: Chi-square tests, comparing Monte Carlo to bootstrap

-	(1) Monte Carlo, 10 reps vs bootstrap, 10 reps	(2) Monte Carlo, 50 reps vs bootstrap, 50 reps
Baseball	4.2	3.4
Boston	8.2* (MC)	4.0
Concrete	13.7** (BS)	2.6
Parkinson's	13.1** (BS)	4.5
Superconductor	16.8*** (BS)	106.1*** (BS)

*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

Finally, the possibility that Monte Carlo and bootstrap require a large number of iterations to perform well was explored. The main study, as presented in Table 1, limits the number of repetitions to a maximum of 50. Section 4.5 addresses the following question:

(5) Which resampling method performed best when performing 1,000 repetitions?

4.1 Increasing reps from 10 to 50

Table 2 shows the Chi-square tests, comparing 10 rep approaches to 50 rep approaches. Looking at Monte Carlo and bootstrap, in all cases except for the Baseball dataset, 50 reps significantly increased performance (in Table 1 compare col 1 to col 5 for Monte Carlo; compare col 2 to col 6 for bootstrap). For the Baseball dataset, performance was about the same between the 10- and 50-iteration approaches. When looking at cross-validation resampling, the trend was also unmistakable: in all cases, the 50 rep approaches (5-fold CV 10 reps and 10-fold CV 5 reps) demonstrated significantly better performance ($p < 0.001$) than their 10 rep counterparts (5-fold CV 2 reps and single-run 10-fold CV, respectively) – in Table 1 compare col 4 to col 8 for 5 fold; compare col 3 to col 9 for 10 fold.

There was one case where using 50 reps was unnecessary: on the Superconductor dataset, perfect performance (100% selection of a good fit λ value) was achieved after using a 10-iteration approach, both 10-fold CV and 5-fold CV 2 reps (see fifth row of Table 1). Hence, we did not run 50-iteration repeated k-fold CV because it would not have resulted in better performance.

For single-run k-fold CV, increasing the number of folds from 10 to 50 did not enhance performance, and, in some cases, performance was significantly worse (this result is discussed in more detail in the next section when considering the effect of k on k-fold CV). This is the one exception where increasing reps from 10 to 50 did not enhance performance.

4.2 Effect of k (number of folds) on k-fold CV

The effect of k can be seen by comparing col 3 to col 7 in Table 1. The Chi-square tests (see col 3 of Table 2) show that increasing k from 10 to 50 folds resulted in worse performance for the Baseball and Boston datasets (these results are indicated by ↓, which signifies that MSE performance moved in the opposite, predicted direction). These results were significant for both the Baseball dataset (decreasing from 30.0% good fit to 23.7% good fit) and the Boston dataset (decreasing from 63.5 to 51.3% good fit). On the other hand, performance was significantly improved on the Concrete dataset (up from 96.5% good fit to 98.5% good fit) and on the Parkinson's dataset (up from 82.3% good fit to 91.6% good fit). These results suggest that increasing the size of k can have mixed results. For larger datasets (in this case, datasets greater than 1,000 rows), performance was improved, but for smaller datasets increasing k beyond 10 folds resulted in poorer performance.

We also compared the performance differences between 5 folds and 10 folds, and the results are displayed in Table 3. For 10-iteration resampling, 5-fold CV 2 reps is compared to 10-fold CV. In all cases, 10-fold CV performs better (see cols 3 and 4 of Table 1), although it is only significant in one case (Boston). For 50-iteration resampling, 5-fold CV 10 reps is compared to 10-fold CV 5 reps. 10-fold CV 5 reps outperformed 5-fold CV 10 reps in all cases (see cols 8 and 9 of Table 1), and these results were significant in two. Hence, the overall recommendation is to use 10 folds rather than 5 folds because 10-fold CV does at least as well and often better than 5-fold CV in all cases studied.

To get a better sense of how the size of k affected performance, we looked more extensively at the Boston dataset, and ran the evaluation on resampling approaches (i.e., running k-fold CV 1,000 times) on a range of k sizes from 10 to 100 folds. Figure 3 graphically displays these results. First, let us focus only on the line labeled "Single-run k-fold." You can see that performance is highest with a k size of 10 and 20, and then it gradually declines for larger values of k. Again, this shows that $k = 10$ is a good choice, and that increasing k beyond 10 for smaller datasets did not result in better performance (the difference in performance between $k = 10$ and $k = 20$ is insignificant). On a related note, we do not believe that LOOCV is ever a good choice for fine tuning a λ parameter in ridge regression. Much better results (with far less computational costs) can be achieved with fewer folds.

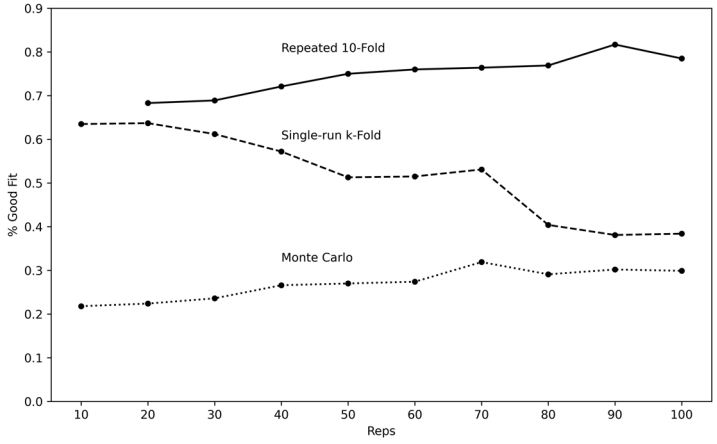

Figure 3: Comparing the performance of repeated 10-fold CV, single-run k-Fold CV, and Monte Carlo on the Boston dataset. The top line (repeated 10 fold) shows how the number of reps increased performance (% good fit) from 2-reps to 10 reps of 10-fold CV on the Boston dataset. The middle line (single-run k-fold) shows how performance decreases as fold size increases from $k =$ 10 folds to $k = 100$ folds. The bottom line (Monte Carlo) shows how performance gradually increases with more reps of Monte Carlo resampling, but in all cases, Monte Carlo is the lowest performing resampling method.

4.3 Single-run CV vs repeated CV

Table 4 shows the Chi-square tests that compare single-run CV to repeated CV (the fifth row of the table, which shows the Superconductor results, can be ignored because 50-iteration repeated CV was not performed). Specifically, two comparisons are made: (1) 50-fold CV (single-run) vs 5-fold CV, 10 reps (repeated) and (2) 50-fold CV (single-run) vs 10-fold CV, 5 reps (repeated). For these comparisons, the analysis is restricted to 50-iteration approaches. For the first comparison, repeated CV performed significantly better on the smaller datasets (i.e., Baseball and Boston), but were statistically insignificant for the larger datasets – see cols 7 and 8 in Table 1. For the second comparison, performance was better in all four cases, significant in three out of the four, and especially pronounced for the smaller datasets – see cols 7 and 9 in Table 1. Again, it is not a coincidence that the smaller datasets, especially, showed more significant improvement using repeated k-fold CV.

To further elucidate the performance advantage of repeated k-fold CV, we looked more extensively, again, at the Boston dataset. Let us return to Figure 3, and compare the performance of repeated k-fold to single-run k-fold; compare the line labeled "Repeated 10 fold" to line labeled "Single-run k-fold. For the first line, we looked at repetitions of 10-fold CV, repeated 2–10 times (i.e., 20–100 reps, in increments of 10). You can see that there is a general increase in performance as you increase the number of reps. Likewise, we looked at single-run CV, where k increases from 10–100 folds, in increments of 10. Here the opposite occurs: the performance of single-run CV trends downwards. Furthermore, in all cases tested, single-run k-fold CV is an inferior performer to repeated k-fold CV.

These results indicate, broadly, that repeated CV is a better performer than single-run CV. The general recommendation is to use 10-fold CV 5 reps, which was the top performer. It appears that this resampling approach will work well across a wide range of dataset sizes and types. For the 10-iteration approaches, on the other hand, 5-fold CV 2 reps did not show any improvement over 10-fold CV – in fact, it declines in performance (compare col 3 to col 4 in Table 1). Because there was no improvement in performance in 5-fold CV 2 reps, we do not display the Chi-square results in Table 4. Hence, the recommendation is to use single-run 10-fold CV if you want a lower (i.e., less computationally intensive) 10-iteration approach.

4.4 Monte Carlo vs bootstrap

Although Monte Carlo and bootstrap are the clear underperformers among the four resampling methods, this study investigates whether there were differences in performance between the two resampling methods. Comparing Monte Carlo to bootstrap is natural because they represent two different, but comparable, ways of randomization; sampling without replacement (in the case of Monte Carlo) vs sampling with replacement (in the case of bootstrap).

Table 5 provides the Chi-square results of the comparison. See also cols 1 vs 2 and 5 vs 6 in Table 1 for the actual results. For 10 reps, bootstrap excels in three cases, and Monte Carlo in one; the fifth case (Baseball) is insignificant. For 50 reps, bootstrap excels in one case (Superconductor), but for the other four cases, the differences are insignificant. Looking at these results, bootstrap has a slight edge over Monte Carlo, but the results are close. Hence, no strong conclusions or recommendations can be made based on this analysis.

4.5 Comparing the resampling methods with a high number of reps

To check that Monte Carlo and bootstrap methods were not disadvantaged by a low number of reps, we conducted a second investigation in which **1,000 reps** of Monte Carlo, bootstrap, and k-fold CV were run. This time, three 1,000-iteration resampling approaches were compared: (1) 1,000 reps of Monte Carlo, (2) 1,000 reps of bootstrap, and (3) 10-fold CV using 100 reps. Like in the main study, we looked at how often the resampling approach chose a good fit, average fit, or poor fit λ value; this was performed 100 times for each of the three approaches. Table 6 also shows the results of the 1,000 iteration approaches. The 50-iteration approaches are also included on this table, and calculated from Table 1, as percentages, for comparison purposes (please note: because we ran the second study only 100 times[1], as opposed to 1,000 times like we did in the main study, percentages are presented in Table 6 in order to make direct comparisons between the two studies).

1 Due to the high computational requirements of running 1,000 reps, the second study was performed only 100 times for each resampling approach.

Table 6: Fit classification results, comparing 50-iteration approaches to 1,000 iteration approaches

	50-iteration approaches			1,000-iteration approaches		
	(1) Monte Carlo 50 reps (%)	(2) Bootstrap 50 reps (%)	(3) 10-fold CV 5 reps (%)	(4) Monte Carlo 1,000 reps (%)	(5) Bootstrap 1,000 reps (%)	(6) 10-fold CV 100 reps (%)
Baseball						
***n* = 263**						
Good fit	21	17	36**	22	14	53**
Average fit	65	67	58	63	61	47
Poor fit	14	16	7	15	25	0
Boston						
***n* = 506**						
Good fit	27	23	75**	56	38	94**
Average fit	63	67	25	44	62	6
Poor fit	10	10	0	0	0	0

**Denotes single best performance.

Only the Baseball and Boston datasets are included in this additional analysis: 1,000 reps were not performed on the other three datasets (i.e., Concrete, Parkinson's, and Superconductor), because their performance levels were approaching near-perfect when using just 50 iterations.

From Table 6, a key finding was that 10-fold CV, run 100 times, was far and away the best performer among the three 1,000-iteration approaches: A chi-square test of independence showed that there was a significant association between resampling method and performance on both the Baseball dataset, ($\chi^2(4) = 55.02$, $p < 0.001$) and the Boston dataset ($\chi^2(2) = 69.87$, $p < 0.001$[2]).

While there was some improvement in performance on the Boston dataset using the 1,000 reps for both Monte Carlo and Bootstrap – e.g., Bootstrap 1,000 reps chose a good fit value 38% of the time vs 23% of the time using 50 reps – the performance was far below that of 10-fold CV 100 reps, which chose a good fit value 94% of the time. In addition, both Monte Carlo and bootstrap did **not** even improve in performance on the Boston dataset using 1,000 reps, the chi-square tests of independence comparing the 50 reps approach to the 1,000 reps approach were not significant for either bootstrap or Monte Carlo. These results show that Monte Carlo and Bootstrap were consistent underperformers, even when performing 1,000 reps.

5 Discussion

Based on the results of this study, three broad recommendations on resampling methods can be made when choosing the right regularization parameter λ in ridge regression modeling (the underlined portion is the broader recommendation).

Recommendation 1: For k-fold CV, the general recommendation is to use $k = 10$ folds; larger values of k can be used on larger datasets. LOOCV is never recommended. For smaller datasets, performance deteriorated with larger numbers of folds (greater than 10). For larger datasets greater than 5,000 samples, 50 folds performed better than 10 folds, but performance with 10 folds was respectable, and achieved at 20% the computational cost. LOOCV (where $k = n$) is never recommended.

Recommendation 2: Repeated k-fold CV is the best overall performer among the four resampling approaches. More specifically, we recommend 10-fold CV 5 reps as a general choice for ridge regression.

2 The chi-square statistic has only two degrees of freedom on the Boston dataset because the poor fit category was eliminated in the calculation: all three of the 1,000-iteration methods chose only good fit and average fit – a poor fit λ value was never chosen.

For larger datasets, the differences in performance are less pronounced between repeated k-fold CV and single-run k-fold CV.

Recommendation 3: Monte Carlo and bootstrap are not recommended as general-purpose resampling methods. These two resampling methods were consistent underperformers in Study 1, across all five datasets studied. In addition, these two resampling methods did not perform better than repeated k-fold CV even when using a large number of reps (i.e., 1,000 reps).

Recommendation 1, which recommends $k = 10$ folds, agrees with conclusions reached by Kohavi [8] and other studies – the consensus in the literature is 5 or 10 folds. Furthermore, as has already been noted in the article, LOOCV is not recommended by many researchers because it results in estimates with high variance, leading to unreliable results. Flach [14] recommends that, as a rule of thumb, a fold should contain a minimum of 30 instances. The recommendation against LOOCV, however, is not universal and unanimous – for example, Kuhn and Johnson [17] do not write it off completely but state that LOOCV can be considered if the number of samples is small. Our recommendation is more direct and straightforward: we never recommend LOOCV, not only because it is computationally inefficient, but also because it provides inferior results compared to repeated k-fold CV. A better way to achieve more accurate estimates of validation error is not to increase the number of folds beyond 10, but to increase the number of repetitions in 10-fold CV.

Recommendations 2 and 3 are largely in line with recent empirical work by Molinaro et al. [31], and Nakatsu [34]. Repeated k-fold CV has been recommended as a way of obtaining more accurate estimates of validation error than single-run k-fold CV. However, these two studies focus exclusively on classification algorithms, not regression modeling, and we are unaware that this result has been demonstrated elsewhere on ridge regression modeling. Furthermore, even though some prior research has demonstrated the effectiveness of repeated k-fold CV, most practitioner books and textbooks do not mention the method – one exception is Kuhn and Johnson [17] who do recommend the method. However, we want to qualify Recommendation 2 by noting that it was not always necessary to increase repetitions. For example, on the larger Superconductor dataset (n = 21,263), it was unnecessary to conduct any repetitions of 10-fold CV at all. For larger datasets, or datasets that quickly converge to a solution, single-run 10-fold CV should suffice.

For Recommendation 3, we have found that Monte Carlo and bootstrap consistently underperform in ridge regression (although in other machine learning algorithms and contexts, including classification algorithms, this may not be the case). Moreover, even though we recognize that Monte Carlo and bootstrap may require a higher number of iterations to perform well, we did not find that they performed better than repeated k-fold CV when using a high number of reps.

Would these results hold and generalize when used on different datasets? To confirm and verify the results of this study, a second independent study was conducted. The results are reported and discussed in a separate appendix, which are available upon request from the authors. In this study, the exact same methodology was used, in which we evaluated nine variations of the four resampling methods. We can report that all three recommendations were verified by the results of the second study. Hence, we are more confident that the recommendations will generalize more broadly, even though we conducted Study 1 on only five datasets.

A key feature of the study's methodology is that it controls for the number of times a ridge regression model was run, to provide baseline comparisons among (1) 10-iteration approaches and (2) 50-iteration approaches. For example, when looking at the effect of fold size on resampling method performance, 10-fold CV 5 reps is compared to 5-fold CV 10 reps (both 50-iteration approaches). Most prior studies simply compare different fold sizes without controlling for overall repetitions (e.g., [31]). Likewise, when comparing single-run CV to repeated CV, appropriate baseline comparisons are performed, for example, 10-fold CV vs 5-fold 2 reps (both 10-iteration approaches) and 50-fold CV vs 10-fold 5 reps (both 50-iteration approaches). We know of few studies that controlled for number of repetitions as systematically and consistently as we did. In the end, we believe that this approach bolsters the final recommendations and conclusions of the study.

Author contributions: The author was responsible for all parts of the article.

Conflict of interest: The author reports there are no conflicting interests to declare.

Data availability statement: The datasets used in this work are available publicity available.

References

[1] Provost F, Fawcett T. Data science for business. Sebastopol, CA: O'Reilly Media; 2013.
[2] Hoerl AE, Kennard RW. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics. 1970;12(1):55–67.
[3] Marquardt DW. Generalized inverses, ridge regression, biased linear estimation and nonlinear estimation. Technometrics. 1970;12(3):591–612.
[4] Nakatsu RT. Information visualizations used to avoid the problem of overfitting in supervised machine learning. International Conference on HCI in Business, Government, and Organizations. Cham, Switzerland: Springer; 2017. p. 373–85.
[5] James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning with applications in R. New York, NY: Springer; 2013.
[6] Burman P. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing method. Biometrika. 1989;76(3):503–14.
[7] Kim JH. Estimating classification error rate: repeated cross-validation, repeated hold-out and bootstrap. Comput Stat Data Anal. 2009;53(11):3735–45.
[8] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. IJCAI. 1995;14(2):1137–45.
[9] Tibshirani RJ, Efron B. An introduction to the bootstrap. Monogr Stat Appl Probab. 1993;57:1–436.
[10] Allen DM. The relationship between variable selection and data augmentation and a method for prediction. Technometrics. 1974;16(1):125–7.
[11] Stone M. Cross-validatory choice and assessment of statistical predictions. J R Stat Soc. 1974;36(2):111–33.
[12] Geisser S. The predictive sample reuse method with applications. J Am Stat Assoc. 1975;70(350):320–8.
[13] Abu-Mostafa YS, Magdon-Ismail M, Lin HT. Learning from data. Vol. 4, New York, NY: AMLBook; 2012.
[14] Flach P. Machine learning: the art and science of algorithms that make sense of data. Cambridge, UK: Cambridge University Press; 2012.
[15] Géron A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. Sebastopol, CA: O'Reilly Media; 2019.
[16] Lantz B. Machine learning with R: expert techniques for predictive modeling. Birmingham, UK: Packt Publishing Ltd; 2019.
[17] Kuhn M, Johnson K. Applied predictive modeling. New York, NY: Springer; 2013.
[18] VanderPlas J. Python data science handbook: essential tools for working with data. Sebastopol, CA: O'Reilly Media; 2016.
[19] Raschka S, Mirjalili V. Python machine learning. Birmingham, UK: Packt Publishing Ltd; 2017.
[20] Breiman L, Spector P. Submodel selection and evaluation in regression. The X-random case. Int Stat Review. 1992;60:291–319.
[21] Efron B. Estimating the error rate of a prediction rule: Improvement on cross-validation. J Am Stat Assoc. 1983;78(382):316–31.
[22] King RD, Orhobor OI, Taylor CC. Cross-validation is safe to use, nature machine intelligence. Psychiatr Res Clin Pract. 2021;3(4):276.
[23] Purushotham S, Tripathy BK. Evaluation of classifier models using stratified ten-fold cross validation techniques. International Conference on Computing and Communication Systems. Berlin: Springer; 2011. p. 680–90.
[24] Raschka S. Model evaluation, model selection, and algorithm selection in machine learning. 2018; arXiv preprint arXiv:1811.12808.
[25] Arlot S, Celisse A. A survey of cross-validation procedures for model selection. Stat Surv. 2010;4:40–79.
[26] Wainer J, Cawley G. Empirical evaluation of resampling procedures for optimising SVM hyperparameters. J Mach Learn Res. 2017;18(15):1–35.
[27] Ghorbani R, Ghousi R. Comparing different resampling methods in predicting Students' performance using machine learning techniques. IEEE Access. 2020;8:67899–911.
[28] Battineni G, Sagaro GG, Nalini C, Amenta F, Tayebati SK. Comparative machine-learning approach: A follow-up study on type 2 diabetes predictions by cross-validation methods. Machines. 2019;7(4):74.
[29] Tamilarasi P, Rani RU. Diagnosis of crime rate against women using *k*-fold cross validation through machine learning. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC). Erode, India: IEEE; 2020. p. 1034–8.
[30] Vakharia V, Gujar R. Prediction of compressive strength and Portland cement composition using cross-validation and feature ranking techniques. Constr Build Mater. 2019;225:292–301.

[31] Molinaro AM, Simon R, Pfeiffer RM. Prediction error estimation: A comparison of resampling methods. Bioinformatics. 2005;21(15):3301–7.
[32] Vanwinckelen G, Blockeel H. On estimating model accuracy with repeated cross-validation. BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning. Ghent, Belgium: Benelearn 2012 Organization Committee; 2012. p. 39–44.
[33] Delaney NJ, Chatterjee S. Use of the bootstrap and cross-validation in ridge regression. J Bus Eco Stat. 1986;4(2):255–62.
[34] Nakatsu RT. An evaluation of four resampling methods used in machine learning classification. IEEE Intell Syst. 2021;36(3):51–7.
[35] Algamal ZY. Shrinkage parameter selection via modified cross-validation approach for ridge regression model. Commun Stat Simul Comput. 2020;49(7):1922–30.
[36] Algamal ZY. A new method for choosing the biasing parameter in ridge estimator for generalized linear model. Chemometr Intell Lab Syst. 2018;183:96–101.
[37] Harrison Jr D, Rubinfeld DL. Hedonic housing prices and the demand for clean air. J Environ Econ Manage. 1978;5:81–102.
[38] Yeh IC. Modeling of strength of high-performance concrete using artificial neural networks. Cem Concr Res. 1998;28(12):1797–808.
[39] Tsanas A, Little MA, McSharry P, Ramig L. Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. IEEE Trans Biomed Eng. 2009;57(4):884–93.
[40] Hamidieh K. A data-driven statistical model for predicting the critical temperature of a superconductor. Comput Mater Sci. 2018;154(3):346–54.

Appendix A

Estimating the λ parameter using Monte Carlo resampling

The intent of this discussion is to show how Monte Carlo resampling reaches identical results to those obtained by repeated k-fold CV, as discussed in Section 3.1 (determining good fit, average fit, and poor fit values). Indeed, given enough repetitions, both the Monte Carlo and bootstrap resampling methods will converge to the same solution, or the same fitting graph.

To illustrate, we estimated the λ value on the Parkinson's dataset. The Parkinson's dataset [39] contains biomedical voice measurements from 42 people with early stage Parkinson's disease ($n = 5{,}875$, $p = 17$ voice measurements). The target variable is Total-UPDRS, which is the Unified Parkinson's Disease Rating Scale, and represents the progression of Parkinson's disease in the patient. Ridge regression was used to fit models on this dataset, using 100 λ values from 0.1 to 1.0. Figure A1 shows the results of average MSE plotted against λ for varying reps of Monte Carlo resampling, 1,000 reps, 4,000 reps, 20,000 reps, and 40,000 reps. You can see that with a larger number of reps, the scatterplot becomes less noisy, and approaches a more discernible fitting graph, or straight-line curve.

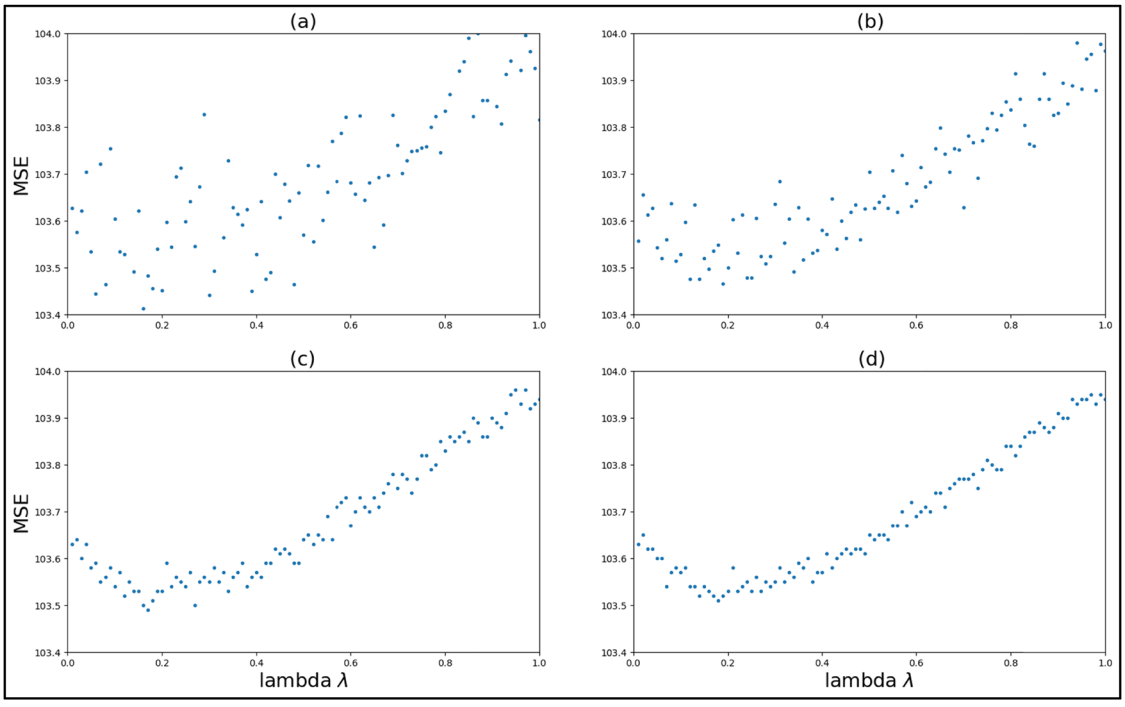

Figure A1: Average MSE over 100 lambda (λ) values from 0.01 to 1.0 on the Parkinson dataset using the Monte Carlo resampling method. The four plots show average MSE, for increasing number of reps: (a) 1,000 reps, (b) 4,000 reps, (c) 20,000 reps, and (d) 40,000 reps.

When only 1,000 reps are used – Figure A1(a) – there is considerable noise in the data, and it is hard to discern an exact fitting graph. By 4,000 and 20,000 reps – Figure A1(b and c) – a clearer picture of the fitting graph emerges. By 40,000 reps – Figure A1(d) – there is only marginal improvement from 20,000 reps, suggesting that there are diminishing returns in running more reps. However, it is clear that the scatterplots are converging to a straight-line curve with more reps.

Figure A2 shows a direct comparison between 2,000 reps of k-fold CV and 40,000 reps of Monte Carlo. You can see that both scatterplots generate similar results. Although we did not run more reps of Monte

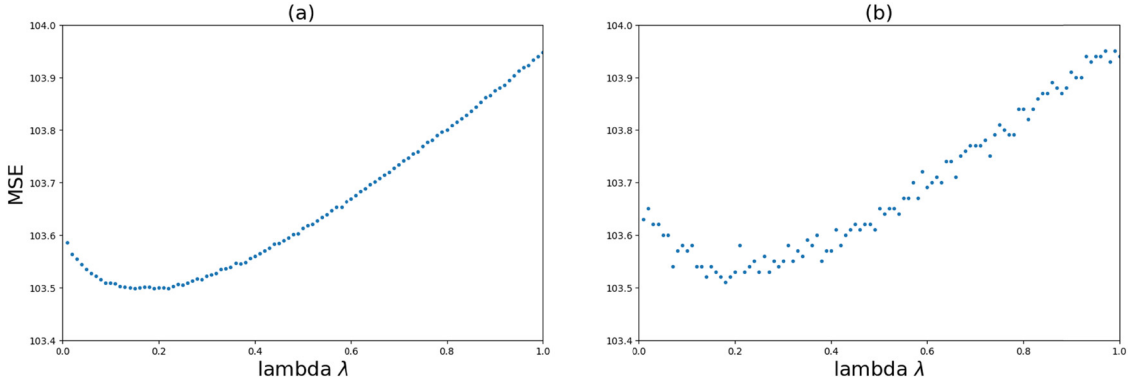

Figure A2: A comparison of the results using (a) 2,000 reps of *k*-fold CV vs (b) 40,000 reps of Monte Carlo.

Carlo, the two approaches – repeated *k*-fold CV and Monte Carlo – will eventually converge to the same solution. As can be seen in this example, repeated *k*-fold CV is much more efficient in coming up with the correct fitting graph with only 2,000 reps, whereas 40,000 reps of Monte Carlo resampling still result in a noisy curve.

Although the results for bootstrap are not reported here, we obtained similar results using the bootstrap resampling method on the Parkinson's dataset. Hence, all three methods – repeated *k*-fold CV, Monte Carlo, and bootstrap – arrive at the same solution, but at different rates.