

Shedding Light on the Role of Sample Sizes and Splitting Proportions in Out-of-Sample Tests: A Monte Carlo Cross-Validation Approach

Christian Janze, Goethe University Frankfurt, Germany, janze@wiwi.uni-frankfurt.de

Abstract

We examine whether the popular 2/3 rule-of-thumb splitting criterion used in out-of-sample evaluation of predictive econometric and machine learning models makes sense. We conduct simulations regarding the predictive performance of the logistic regression and decision tree algorithm when considering varying splitting points as well as sample sizes. Our non-exhaustive repeated random sub-sampling simulation approach known as Monte Carlo cross-validation indicates that while the 2/3 rule-of-thumb works, there is a spectrum of different splitting proportions that yield equally compelling results. Furthermore, our results indicate that the size of the complete sample has little impact on the applicability of the 2/3 rule-of-thumb. However, our analysis reveals that when considering relatively small and relatively large training samples in relation to the sample size, the variation of the predictive accuracy can lead to misleading results. Our results are especially important for IS researchers considering the usage of out-of-sample methods for evaluating their predictive models.

Keywords: Out-of-Sample Testing; Monte Carlo Cross-Validation; 2/3 Rule-of-thumb; Logistic Regression; Decision Tree Algorithm

1. INTRODUCTION

Since the publication of a landmark paper on the lack of predictive modeling in Information Systems (IS) research (Shmueli & Koppius, 2011), scholars introduced sophisticated econometric and machine learning methods to the field. Researchers routinely create more than one predictive model while hunting for a particularly well performing one. A substantial body of research is therefore concerned with the inevitable model selection problem while developing prediction models (Zhang & Yang, 2015). To select a model, researchers can calculate and compare the predicted outcome with the actual outcome using the data that was used to *fit* or *train* the model (terminology varies between econometrics and machine learning). However, it is agreed upon that "significant in-sample evidence of predictability does not guarantee significant out-of-sample predictability" (Inoue & Kilian, 2005).

To increase the understanding of issues related to in-sample validation techniques, let's consider an entrepreneur on a budget that is interested in the question on whom to target in a postal mailing campaign. There are two possible reactions from recipients to his mailing: respond or not respond. He decides to utilize data of his previous campaigns to maximize the utility of his efforts. Specifically, he tries to identify factors (e.g. age, gender, income, etc.) influencing the recipients'

responsiveness. Drawing on his econometric- and machine learning knowledge, he fits a logistic regression to the data and utilizes the C5.0 algorithm to grow a decision tree. Thrilled by the fact that his models achieve an almost perfect predictive accuracy on the historic data, he spends his entire marketing budget on recipients selected by his models. However, a few weeks later, almost no one has responded. How is this possible? A likely answer is that his models are extremely specific, lack generalizability and are thus subject to the overfitting problem.

To account for the overfitting problem, researchers came up with the idea of out-of-sample testing (aka hold-out-method) (Schneider, 1997). Out-of-sample testing is used to evaluate and subsequently compare the predictive performance of models on new and from the perspective of the model "unseen" data. This means that the data sample is split into sub-samples. While the first sub-sample is used to fit/train a model, the second one is used to test it. Thus, these sub-samples are often referred to as the training- and test samples. However, how could someone decide between - say splitting the data sample into equally sized sub-samples and the oftentimes used 2/3 rule-of-thumb (Cios, Pedrycz, Swiniarski, & Kurgan, 2007; Dobbin & Simon, 2011) of using 2/3 of the data for training and the remaining 1/3 for testing purposes? Thus, we state our two research questions as follows:

- *Does the popular 2/3 rule-of-thumb splitting criterion used in out-of-sample tests make sense?*
- *Does the applicability of the 2/3 rule-of-thumb depend on the initial sample size?*

We operationalize both research questions by means of multiple simulations covering both the logistic regression as well as the C5.0 decision tree algorithm as representatives for popular econometric and machine learning approaches used in binary outcome predictions.

To find an answer to our first research question, we repeatedly train both the logistic regression and the C5.0 decision tree algorithm on varying splitting proportions of randomized samples and calculate different performance metrics. In other words, we step-wise increase the size of the training sample and thus decrease the size of the test sample. For an assessment of our second research question, we not only vary the splitting proportion but also the initial sample size. We find that while the 2/3 rule-of-thumb works well on average, there is a whole spectrum of different splitting proportions that yield equally compelling results. Furthermore, our results indicate that the size of the complete sample has no considerable impact on the applicability of the 2/3 rule-of-thumb. However, our analysis shows that when considering relatively small and relatively large training samples in relation to the initial sample size, the variation of the predictive accuracy can lead to situations in which out-of-sample tests yield misleading results.

The remaining portion of this paper is structured as follows: First, we provide background on cross-validation methods and out-of-sample tests. Furthermore, we provide an overview on

measures of fit. Second, we outline our research methodology and two-step simulation design. Third, we summarize the data set used in our study as well as the model specification. Fourth, we present the result of our simulations. Lastly, we provide a discussion and a conclusion of the study.

2. BACKGROUND

2.1. Cross-Validation and Out-of-Sample Tests

Zhang and Yang (2015) point out that cross-validation (see Allen, 1974; Stone, 1974; Geisser, 1975) is a technique to evaluate the predictive performance of a model. Cross-validation methods can be classified into *exhaustive* (e.g. leave-one-out) as well as *non-exhaustive* (e.g. k-fold cross-validation, 2-fold cross-validation and repeated random sub-sampling) approaches.

Monte Carlo cross-validation (Picard and Cook, 1984), is a representative of a non-exhaustive repeated random sub-sampling method (Dubitzky, Granzow, & Berrar, 2007). In comparison to other model selection methods (bootstrapping, Akaike information criterion, etc.), Monte Carlo cross-validation is asymptotically consistent, meaning its predictive ability converges to 1 as the total number of observations n approaches infinity (Shao, 1993). The basic procedure consists of three steps as discussed by Kaynak, Alpaydin, Oja and Xu (2003): first, elements of a data sample X are chosen at random without replacement to create a new training data sample X_1 . The remaining elements of X are used to form the test data sample X_2 . Second, the model M is trained with X_1 . Subsequently, the mean squared error is calculated by subtracting the actual values of the elements in X_2 from their predictive values obtained by computing $M(X_2)$ and averaging the squared results. Finally, the first two steps are repeated many times and the arithmetic mean is computed to yield the average error.

Out-of-sample tests are a special case of cross-validation (Schneider, 1997), namely a 2-fold-cross validation. Furthermore, it is a special case of the Monte Carlo cross-validation that lacks the third step described above (Kaynak et al., 2003). Obviously, out-of-sample tests require the researcher to split the data into sub-samples - the training sample used to train the model and the test sample. However, it is a challenging question what splitting proportion a researcher should use. Thus, many researchers use heuristics such as the popular 2/3 rule-of-thumb (see for example Cios, Pedrycz, Swiniarski and Kurgan, 2007; Dobbin and Simon, 2011), suggesting to use 2/3 of the data sample for training and the remaining 1/3 for testing purposes.

To the best of our knowledge, no previous study examined the appropriateness of the 2/3 rule-of-thumb splitting criterion in a binary dependent variable setting by means of repeated random sub-sampling procedures (i.e. Monte Carlo cross-validation) of both econometric (logistic regression) and machine learning (C5.0 decision tree algorithm) approaches. We therefore proceed to describe

various measures of fit we use in the following sections to examine the overall performance of predictive models depending on the splitting proportion and initial sample size.

2.2. Confusion Matrix and Measures of Fit

Table 1 illustrates possible predicted/actual outcome combinations of the binary outcome problem the entrepreneur of our initial example faces in his mailing campaign where recipients can either respond or not respond. True positives (TP) refer to cases in which a potential customer reached out to the entrepreneur as predicted by the model whereas true negatives (TN) are cases in which the models correctly predicted a non-responding potential customer. False negatives (FN) mean that the model predicted that the customer would not respond but did. Thus, FN represent cases of potentially missed business opportunities. The last possible outcome are false positives (FP) in which a model predicts the response of a recipient that does not respond. This is a costly mistake because the money spent on sending the mail is wasted.

		PREDICTED RESPONSE	
		Yes	No
ACTUAL RESPONSE	Yes	True Positive (TP)	False Negatives (FN)
	No	False Positives (FP)	True Negatives (TN)

Table 1 - Confusion Matrix (Adapted from Lantz, 2015)

As discussed by Lantz (2015), these combinations of predicted and actual outcomes can be used to calculate various model performance metrics from which we outline commonly used ones: first, the accuracy (Equation 1), which represents the overall share of correctly classified examples (positive and negative outcomes) and second (Equation 2), the error rate, which is the "proportion of incorrectly classified examples" (Lantz, 2015). Third, the sensitivity (Equation 3), describing the "proportion of positive examples that were correctly classified" (Lantz, 2015). Fourth, the specificity (Equation 4), which is the "proportion of negative examples that were correctly classified" (Lantz, 2015).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (3)$$

$$Error Rate = \frac{FP + FN}{TP + TN + FP + FN} = 1 - accuracy \quad (2)$$

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

3. RESEARCH METHODOLOGY AND SIMULATION DESIGN

To investigate our two research questions, we rely on a non-exhaustive cross-validation approach, which is based on repeated random sub-sampling known as Monte Carlo cross-validation (Dubitzky et al., 2007). We do so for both the logistic regression and the decision tree algorithm.

In the following, we describe our two-step approach to tackle both research questions consecutively.

3.1. Design of Step 1: Vary Splitting Proportions

In research question one we are interested in whether the popular 2/3 rule-of-thumb thumb used as a splitting criterion in out-of-sample testing makes sense. To shed light on this question, it seems natural to compare the achieved out-of-sample predictive accuracy depending on the splitting point. Assuming the rule-of-thumb is a useful heuristic, one would expect to detect a local optimum evaluative metric (e.g. a high predictive accuracy) while exploring the simulation results.

We operationalize this as stylized in Figure 1. First, we randomize the row order of the full data sample. Next, we run both the logistic regression and decision tree algorithm for step-wise increased splitting points (i.e. increasingly large training samples). Subsequently, we calculate the evaluation metrics (accuracy, error rate, sensitivity and specificity). We repeat these first three steps 100 times (arbitrarily chosen) to rule out potential biases because of the ordering of the data sample. Finally, we aggregate and visualize the results.

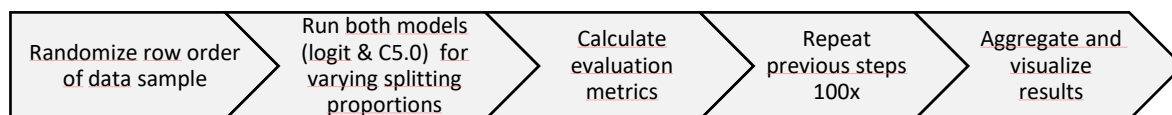


Figure 1 - Stylized Operationalization of Research Question One

Our R implementation of this flow chart is shown in Table 2. While the outer loop (ignore the bold face loop for now) describes the repetition of the entire procedure for 100 times, its body includes the randomization of the full data sample (*dat*) and subsequent assignment to the *sdat* variable (the *dat* variable itself remains unchanged). The following nested inner-loop proceeds to step-wise increase the size of the training-sample by 100. We do so to reduce the computational resources required to execute the program. For each of these splits, we calculate confusion matrices for both the logistic regression as well as the decision tree algorithm using the `runLogit` and `runDTree` convenience functions (see Appendix). We then store the results in the *results* matrix.

```

for (z in 200:2500, by 100) {
  for (k in 1:100) {
    #Randomize row order of data sample dat
    sdat<- dat[sample(nrow(dat)),]
    for(i in seq(100,(nrow(dat)-100), 100)) {
      #Split sdat in train and test
      train <- sdat[1:i,]
      test <- sdat[(i+1):nrow(dat),]
      # Calculate confusion matrices for current split
      lc <- runLogit(mdl.form,train,test) # returns logistic regression confusion matrix
      dc <- runDTree(mdl.form,train,test) # returns decision tree confusion matrix
      # Store results of current iteration
      results <- rbind(results, c(k, i, 1, lc[1,1], lc[1,2], lc[2,1], lc[2,2]))
      results <- rbind(results, c(k, i, 2, dc[1,1], dc[1,2], dc[2,1], dc[2,2]))
    }
  }
}

```

Table 2 - R Implementation of Monte Carlo Cross-Validation Variant (*Note: Bold-faced Outer-loop is Only Applied in Step 2 of the Analysis*)

3.2. Design of Step 2: Vary Splitting Proportions and Initial Sample Sizes

In our second research question, we examine the role of the overall sample size in out-of-sample testing. We do so by not only varying the splitting proportions as shown in the previous section but also the initial sample size. As shown in Figure 2, the only difference to our operationalization of our analysis regarding the first research question depicted in Figure 1 is the first step in which we vary the initial sample size.

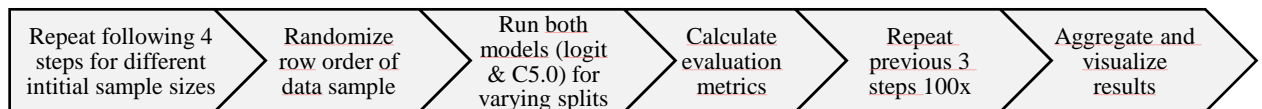


Figure 2 - Stylized Operationalization of Research Question Two

To allow for the examination of the impact of different initial sample sizes used when considering out-of-sample tests applying the 2/3 rule-of-thumb as the splitting criterion, we added another loop to the script as presented in Table 2. This loop step-wise increases the initial sample size, starting from 200 elements to a maximum of 2,500 in steps of 100.

4. DATASET AND MODEL SPECIFICATION

In our study, we use a data set provided by a large German bank regarding a mailing campaign. The data set contains a total of 32 variables and 8,000 equally weighted observations including a binary variable representing the response of a recipients of the postal mailing. However, as we are more interested in studying the behavior of out-of-sample tests with regards to varying splitting proportions and sample sizes, we did not further explore the specific data at hand. To streamline our analysis, we removed all incomplete cases and very sparse features from our analysis. Our final data set used in the following simulations contains a total of 29 variables and 6,928 complete

observations. To specify both models, we used the response variable as the dependent variable of the logistic regression and as a target feature of the decision tree algorithm. The remaining variables are used as explanatory variables/features.

5. SIMULATION RESULTS

5.1. Results of Step 1: Varying Splitting Proportions

Descriptive statistics of our results of step 1 are shown in Table 3. In total, we fitted/trained the logistic regression and decision tree 6,800 times using the training portion of the data sample. The results presented in the table represent measures of fit calculated using the respective testing samples.

In the following, we summarize the most noteworthy observations: First, both the logistic regression and decision tree yield a high variability in any of the four performance measures. For example, in case of the logistic regression, the accuracy ranges from as low as 0.5 to as high as 0.9. The same can be observed for the error rate, sensitivity and specificity. This indicates that out-of-sample predictions react quite significantly to varying splitting proportions. Second, our results show that the decision tree algorithm outperforms the logistic regression in all tested performance metrics (accuracy, error rate, sensitivity and specificity). For example, the mean accuracy of 81% of the logistic regression is significantly lower compared to the 87% of the decision tree. Furthermore, the logistic regression achieved a range of predictive accuracy of 0.39 and thus almost doubles the range of the decision tree algorithm of 0.2.

METHOD	METRIC	RUNS	MEAN	STD. DEV.	MEDIAN	MAD	MIN	MAX	RANGE
Logistic Regression	Accuracy	6,800	0.81	0.02	0.81	0.01	0.50	0.90	0.39
	Error Rate	6,800	0.19	0.02	0.19	0.01	0.10	0.50	0.39
	Sensitivity	6,800	0.82	0.03	0.82	0.02	0.56	0.95	0.39
	Specificity	6,800	0.80	0.02	0.80	0.02	0.43	0.91	0.47
Decision Tree Algorithm	Accuracy	6,800	0.87	0.01	0.87	0.01	0.73	0.93	0.20
	Error Rate	6,800	0.13	0.01	0.13	0.01	0.07	0.27	0.20
	Sensitivity	6,800	0.84	0.03	0.84	0.02	0.60	0.94	0.34
	Specificity	6,800	0.90	0.03	0.90	0.03	0.67	0.98	0.30
Delta %	Accuracy	-	-6.90%	100.00%	-6.90%	0.00%	-31.51%	-3.23%	95.00%
	Error Rate	-	46.15%	100.00%	46.15%	0.00%	42.86%	85.19%	95.00%
	Sensitivity	-	-2.38%	0.00%	-2.38%	0.00%	-6-67%	1.06%	14.71%
	Specificity	-	-11.11%	-33.33%	-11.11%	-33.33%	-35.82%	-7.14%	56.67%

Table 3 - Descriptive Statistics of Step 1 (Varying Splitting Proportions)

We now proceed to conduct a visual analysis of the results of step 1. Figure 1 visually depicts the mean accuracy, the mean error rate, the mean sensitivity and mean specificity achieved by the logistic regression model and decision tree algorithm for each splitting point. Most strikingly, we observed that both the logistic regression and the decision tree algorithm converge relatively fast to

their highest mean accuracy and lowest error rate: Around a splitting point of 2,000, which means in this case a splitting proportion of around 30% of the data in the test sample and 70% in the training sample, the predictive accuracy (and thus the error rate) doesn't change much. Nevertheless, the 2/3 rule-of-thumb in question appears to work quite well.

However, we assume that the variability of the results will increase largely when using extreme splitting proportions with only very few observations within the training or test sample. To examine this, we will conduct an additional analysis in step 2. Nevertheless, it is an interesting result that the predictive accuracy metric doesn't drop more harshly when using almost all the data available to train the model. Another noteworthy observation is that there is no instance in which the decision tree algorithm doesn't dominate the logistic regression model.

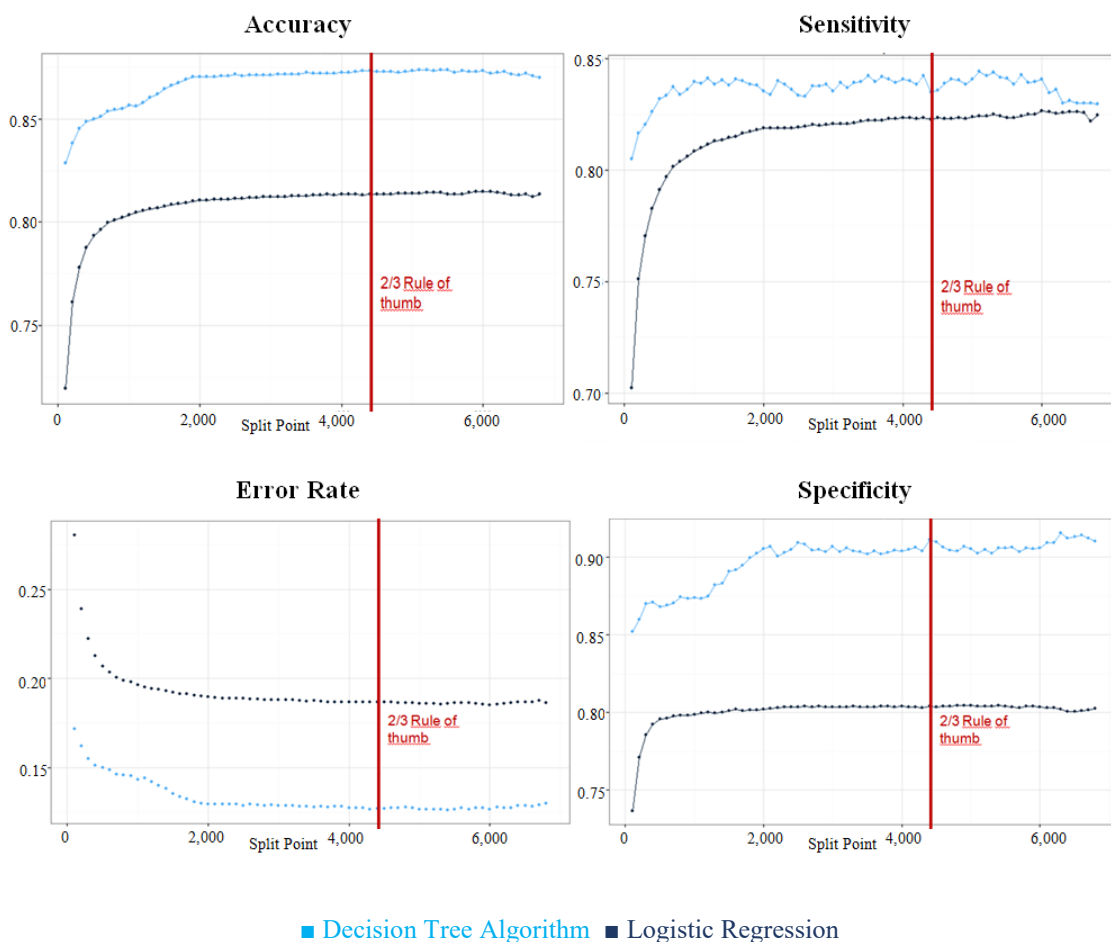


Figure 3 - Mean Accuracy, Mean Error Rate, Mean Sensitivity and Mean Specificity (Y-Axis) of Logistic Regression and Decision Tree Algorithm Depending on Splitting Proportions (X-Axis) (Note: Values on Vertical Axis Represent Mean of 100 Runs)

Regarding the mean sensitivity and mean specificity, it is evident that the fast convergence of both metrics in case of the mean accuracy and mean error rate holds true. Furthermore, in terms of the specificity, the logistic regression reaches its "steady state" a lot faster than the decision tree algorithm. Another observation is the relatively high variance of the decision trees' mean

sensitivity and specificity measures when compared to the logistic regression. This could be due to outliers hidden in the means that yield especially high or low sensitivity and specificity measures. In summary, our results of step 1 - in which we varied only the splitting proportions applied to an otherwise fixed sample size - include three main findings: First, our results indicate that the rule-of-thumb works quite well. Nevertheless, there is an entire range of other and probably equally good rules-of-thumbs that yield the same result. Second, it is surprising that once a certain threshold of examples in the training sample is reached, all mean performance measures included in our analysis converge quite fast and do not drop when the training sample becomes very large. Third, the decision tree algorithm outperforms the logistic regression in this examination on every level. No matter whether considering the accuracy, error rate, sensitivity or specificity.

Our analysis so far has clearly shown that splitting proportions have an impact on performance measures widely used. In the next section, we will specifically consider sample sizes up to 2,500 observations. This is mostly because our results so far indicate that both the logistic regression and decision tree algorithm converges relatively fast. We are interested in what happens when the sample size is decreased and furthermore, how the models' convergence looks when considering smaller data sets.

5.2. Results of Step 2: Varying Splitting Proportions and Initial Sample Sizes

In the following, we present the results of step 2 of our analysis, an assessment of the behavior of out-of-sample predictions when varying both the sample size and splitting proportions. Figure 4 presents a three-dimensional visualization of our results. Each point represents the mean accuracy of the logistic regression function depending on the initial sample size and splitting proportion. It is clear that very small training samples (indicated by a small splitting point) yield low levels of mean accuracy.

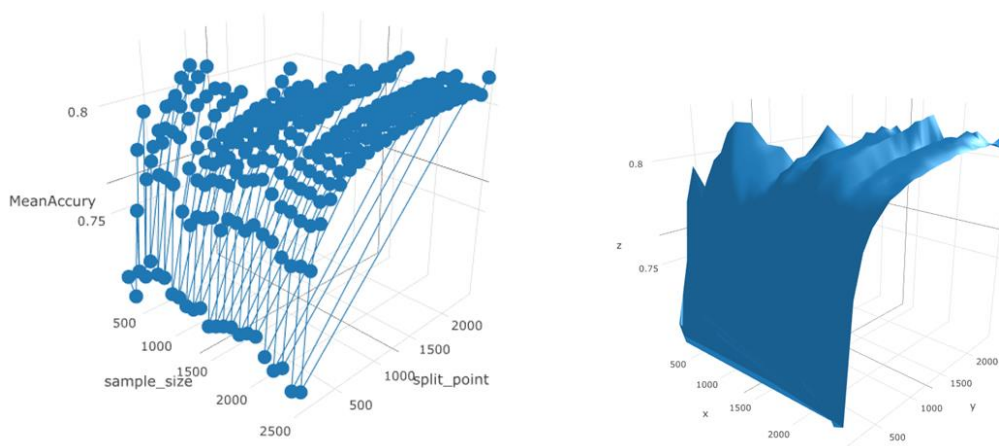


Figure 4 - Three-dimensional Visualization of the Mean Accuracy (Z-Axis) of the Logistic regression depending on Splitting Points (Y-Axis) and Sample Sizes (X-Axis)

Figure 5 presents the same data but on a two-dimensional plane. These charts visualize the mean accuracy and mean error rate of the logistic regression depending on the sample size and splitting proportion applied. The colors areas in this type of chart can be interpreted as horizontal slices through the "mountain" presented in Figure 4. The higher the mean accuracy or mean error rate, the lighter the area.

The first finding is that the 2/3 rule-of-thumb, which is represented by the dashed red line plotted on top of the chart, works well. This is because the dashed line lies to a great extent on light (i.e. more yellow) areas when considering the mean predictive accuracy. Likewise, the mean error rate plot shows that the 2/3 rule-of-thumb line lies to most parts on dark, i.e. mostly blue, areas. The same observation holds true for both the mean specificity and mean sensitivity of the logistic regression and decision tree as shown in Figure 7 (Appendix).

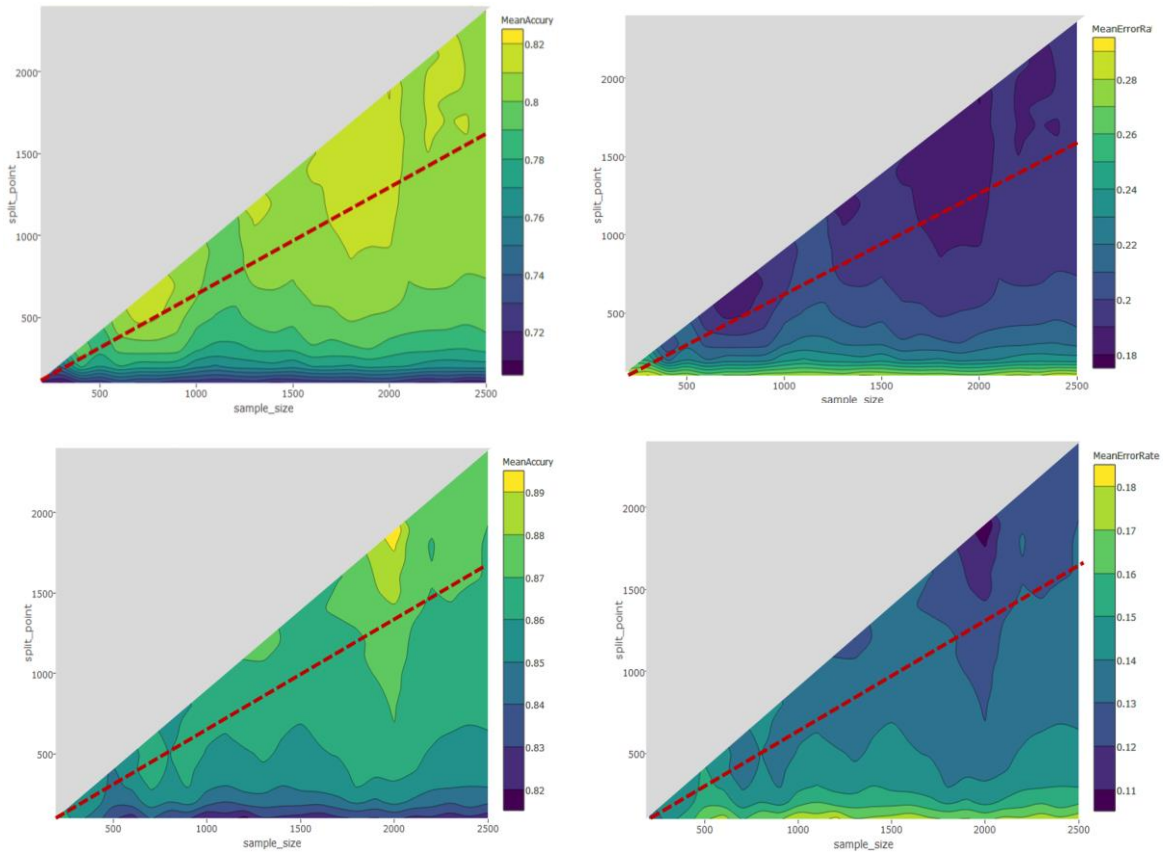


Figure 5 - Mean Accuracy (Left) and Mean Error Rate (Right) of the Logistic Regression Model (Top) and Decision Tree (Bottom) Depending on the Sample Size (X-Axis) and Splitting Proportion (Y-Axis)
(Note: Dashed Red Line Represents 2/3 Rule-of-thumb)

Previously in this work, we suspected that the variation within our analysis increases significantly when considering cases in which the training sample is overly large or small and that the fast convergence of our results to a stable point is mostly due to averaging. In the following we test this by comparing the mean predictive accuracy with its standard deviation of both the logistic

regression model and decision tree algorithm depending on the splitting proportion chosen for initial sample sizes of 500, 100 and 1,500.

Figure 6 presents the mean accuracy and standard deviation of the logistic regression for three different sample sizes and varying splitting proportions. We see that our assumption pointed into the right direction. The plots clearly indicate that at the tails, i.e. very small and very large training samples relative to the total sample size, the standard deviation increases. This indicates that our previous results of stable performance evaluation metrics and their quick convergence was largely due to averaging the results. The same pattern holds true for the decision tree algorithm as shown in Figure 8 (Appendix).

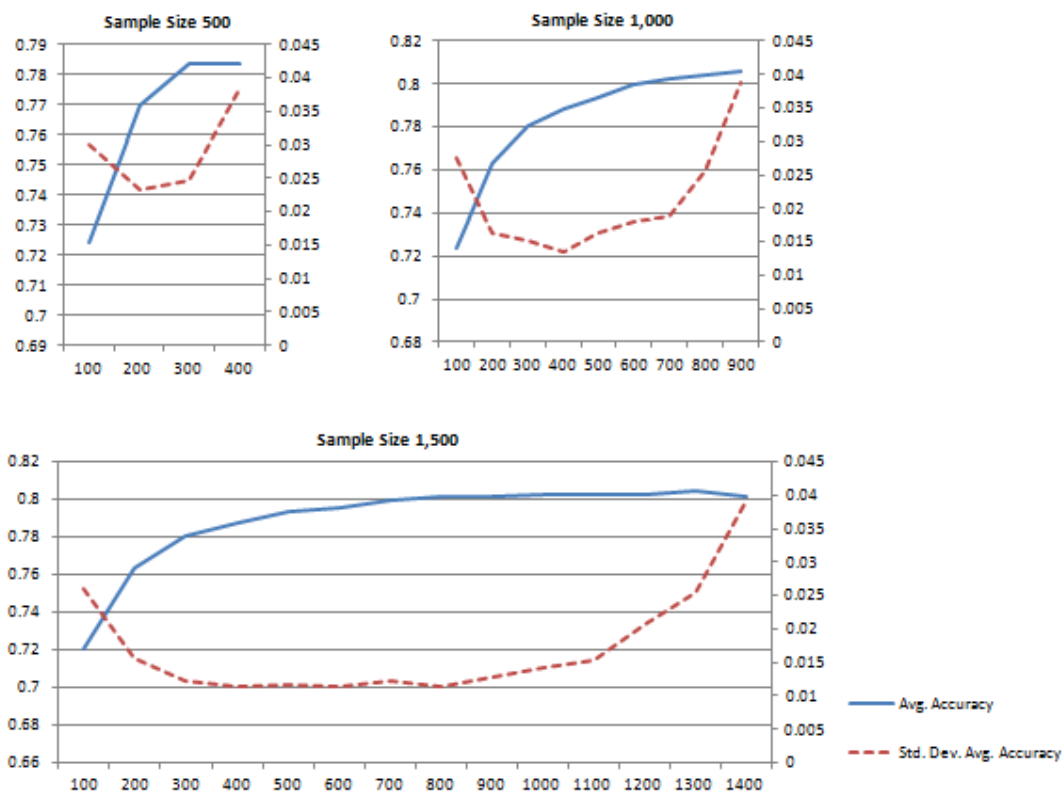


Figure 6 - Mean accuracy (left axis) and standard deviation of the mean accuracy (right axis) of the logistic regression depending on splitting points and the sample sizes

The results of step 2 of our analysis revealed the following two noteworthy findings: First, considering the mean performance measures alone (accuracy, error rate, sensitivity and specificity), the 2/3 rule of thumb appears to provide stable results for samples of varying size. Second, a closer examination of the results revealed that the fast convergence of the performance metrics to a stable level once a certain splitting proportion (i.e. size of the training sample) is reached, is largely due to averaging the results.

6. DISCUSSION AND CONCLUSION

Faced with the possibility to create a large number of predictive models, researchers depend on model selection mechanisms (Zhang & Yang, 2015). It is widely agreed upon that "significant in-sample evidence of predictability does not guarantee significant out-of-sample predictability" (Inoue & Kilian, 2005). This is mostly because of the problem of overfitting, meaning that a model "has captured both random noise as well as genuine nonlinearities" (Campbell, Lo, & MacKinlay, 2012). To tackle this problem, many researchers rely on out-of-sample validations, which is also known as the hold-out-method validation (Schneider, 1997). At its core, out-of-sample validations split a dataset into a training- and test sample. While the former is used to train the model, the latter is used to evaluate the performance of the model.

However, the question whether the frequently applied rule-of-thumb of using 2/3 of the data for training and the remaining 1/3 for testing purposes (Cios et al., 2007; Dobbin & Simon, 2011) makes sense and whether its applicability depends on the sample size often remains an open question. Because of this, we stated two research questions: First, does the popular 2/3 rule-of-thumb splitting criterion used in out-of-sample tests generally make sense? Second, does the applicability of the 2/3 rule-of-thumb depend on the initial sample size?

To shed light on these question, we relied on different simulations of the predictive performance of the logistic regression as well as the C5.0 decision tree algorithm. We used a real world data sample provided by one of the largest German banks. The data sample covers a multitude of different socio-economical variables and the response to a marketing campaign. Our analysis consists of two steps. To examine our first research question, we varied the splitting proportions while holding the initial sample size fixed to study the behavior of out-of-sample tests. In the second step of our analysis, we varied the sample size as well as the splitting proportion. For operationalization, we designed a two-step simulation study which shares many characteristics with a technique known as Monte Carlo cross-validation (Picard and Cook 1984).

The results of our first step simulation design show that the 2/3 rule-of-thumb works quite well but indicate that there is an entire range of other and probably equally suitable splitting proportions that yield similar results. Second, we show that once a certain threshold of examples in the training sample is reached, all mean performance measures included in our analysis converge fast and even more interestingly do not drop when the training sample becomes very large. Third, the C 5.0 decision tree algorithm outperforms the logistic regression in every aspect tested. Our second step simulation design yields two additional important insights regarding the behavior of out-of-sample tests when varying both the sample size and splitting points: First, the 2/3 rule-of-thumb works well when considering varying initial sample sizes. Second, the variation of the predictive

accuracy increases significantly when the training sample is either very small or very large relative to the initial sample size.

Therefore and in summary, the answer to our first research question is that while the 2/3 rule-of-thumb works well on average, there is a whole spectrum of different splitting proportions that yield equally compelling results. Furthermore, our results indicate that the initial sample size has little impact on the applicability of the 2/3 rule-of-thumb. Nevertheless, researchers must be very careful when interpreting their results: Our analysis reveals that when considering relatively small and relatively large training samples in relation to the initial sample size, the variation of the predictive accuracy can lead to situations in which the out-of-sample tests yield misleading results. For example, depending on the row ordering of the data sample, it is possible that an out-of-sample based evaluation of the predictive performance of a model yields tempting results which are - unfortunately - spurious.

We advise researchers and practitioners alike not to base their model evaluation solely on a single out-of-sample test. One feasible method to overcome many of the problems of out-of-sample tests was used in this study: Monte Carlo cross-validation.

REFERENCES

- Allen, D. M. (1974). The Relationship Between Variable Selection and Data Augmentation and a Method for Prediction. *Technometrics*, 16(1), 125–127. <https://doi.org/10.1080/00401706.1974.10489157>
- Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (2012). *The Econometrics of Financial Markets* (2nd ed.). Princeton University Press.
- Cios, K. J., Pedrycz, W., Swiniarski, R. W., & Kurgan, L. (2007). *Data Mining: A Knowledge Discovery Approach*. Springer Science & Business Media.
- Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *BMC Medical Genomics*, 4(1), 1.
- Dubitzky, W., Granzow, M., & Berrar, D. P. (2007). *Fundamentals of Data Mining in Genomics and Proteomics*. Springer Science & Business Media.
- Geisser, S. (1975). The Predictive Sample Reuse Method with Applications. *Journal of the American Statistical Association*, 70(350), 320–328. <https://doi.org/10.1080/01621459.1975.10479865>
- Inoue, A., & Kilian, L. (2005). In-Sample or Out-of-Sample Tests of Predictability: Which One Should We Use? *Econometric Reviews*, 23(4), 371–402. <https://doi.org/10.1081/ETC-200040785>
- Kaynak, O., Alpaydin, E., Oja, E., & Xu, L. (2003). *Artificial Neural Networks and Neural Information Processing — ICANN/ICONIP 2003: Joint International Conference ICANN/ICONIP 2003, Istanbul, Turkey, June 26–29, 2003, Proceedings*. Springer.
- Lantz, B. (2015). *Machine Learning with R - Second Edition: Amazon.de: Brett Lantz: Fremdsprachige Bücher*. Packt Publishing. Retrieved from <http://www.amazon.de/Machine-Learning-R-Second-Edition/dp/1784393908>
- Picard, R. R., & Cook, R. D. (1984). Cross-Validation of Regression Models. *Journal of the American Statistical Association*, 79(387), 575–583. <https://doi.org/10.2307/2288403>
- Schneider, J. (1997, February 7). Cross Validation. Retrieved March 30, 2016, from <http://www.cs.cmu.edu/~schneide/tut5/node42.html>
- Shao, J. (1993). Linear Model Selection by Cross-Validation. *Journal of the American Statistical Association*, 88(422), 486. <https://doi.org/10.2307/2290328>
- Shmueli, G., & Koppius, O. R. (2011). Predictive analytics in information systems research. *Mis Quarterly*, 553–572.

Stone, M. (1974). Cross-validators choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 111–147.

Zhang, Y., & Yang, Y. (2015). Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, 187(1), 95–112. <https://doi.org/10.1016/j.jeconom.2015.02.006>

APPENDIX

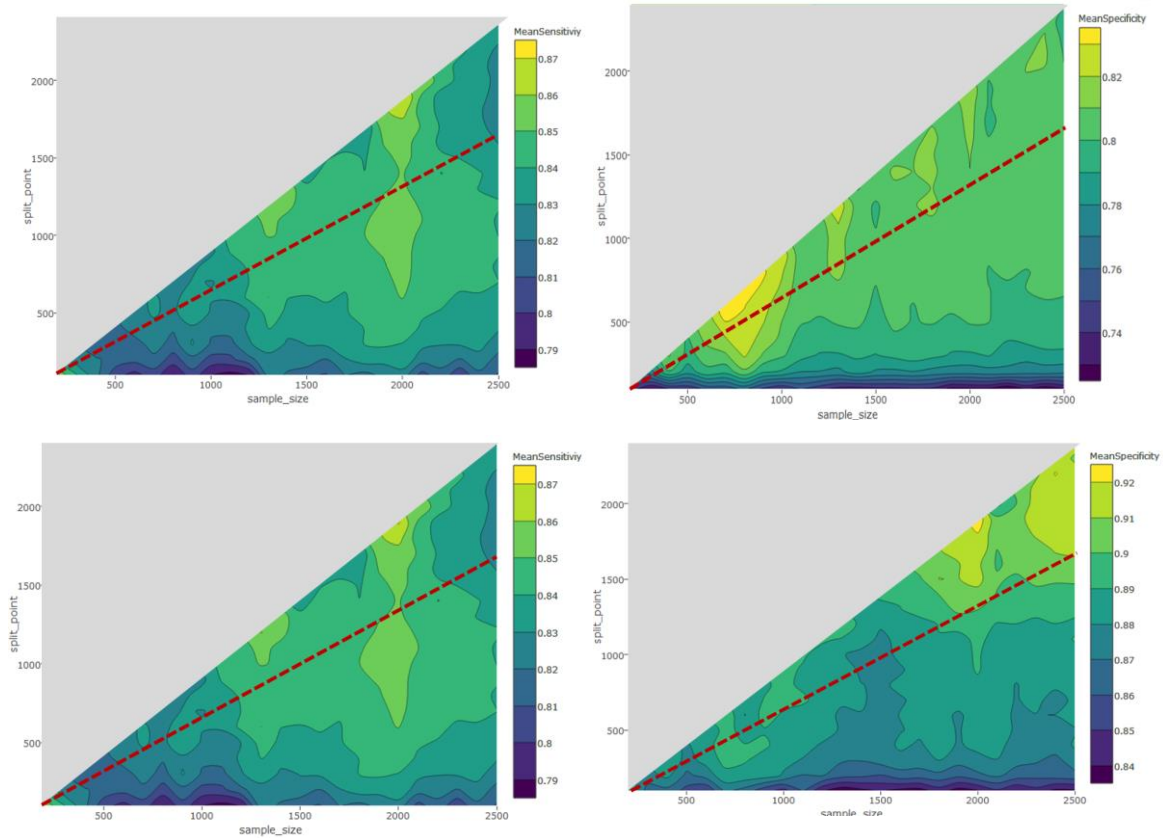


Figure 7 - Mean Sensitivity (Left) and Mean Specificity (Right) of the Logistic Regression (Top) and Decision Tree Algorithm (Bottom) depending on Sample Size and Splitting Proportion
(Note: Dashed Line Represents 2/3 Rule-of-Thumb)

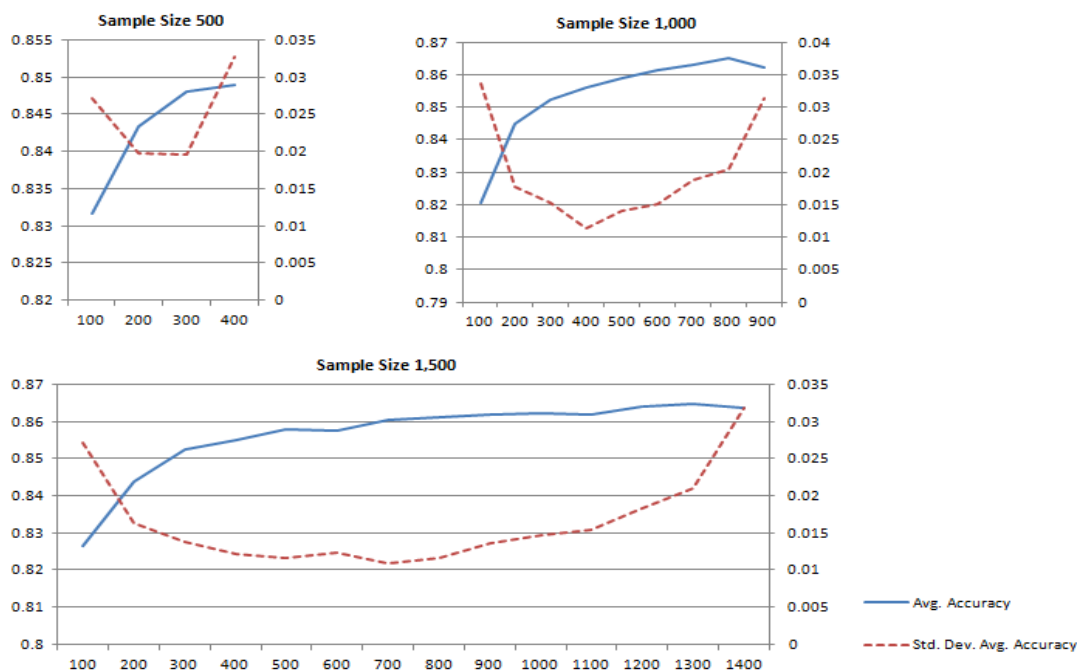


Figure 8 - Mean Accuracy (Left Axis) and Standard Deviation of Mean Accuracy (Right Axis) of the Decision Tree Algorithm depending on Splitting Proportions

```
xtable <- function(vactual, vpredicted) {
  return(CrossTable(vactual, vpredicted,
    prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
    dnn = c('actual', 'predicted')))
```

Table 4 - R Function to Calculate Confusion Matrix

```
runLogit <- function(model, training, test){
  #Fit logit model to train data
  res.lgt <- glm(mdl.form, family=binomial(link="logit"), data = train)
  # Predict binary outcome using res.lgt with test sample
  lgt.pr <- predict(res.lgt,newdata=test,type="response")
  lgt.pr <- round(lgt.pr)
  #Evaluate model performance
  conf<-xtable(test$response, lgt.pr)
  #return confusion matrix
  return(conf$t) }
```

Table 5 - R Function to Return Confusion Matrix of Logistic Regression

```
runDTree <- function(model, training, test){
  # Grow decision tree using train data
  dtree1 <- C5.0(mdl.form, train)
  #calculate predicted outcome with test data
  dtree.pr <- predict(dtree1, test[-1])
  #evaluate model performance
  conf<- xtable(test$response, dtree.pr)
  #return confusion matrix
  return(conf$t) }
```

Table 6 - R Function to Return Confusion Matrix of Decision Tree Algorithm