sciendo

The Proportion for Splitting Data into Training and Test Set for the Bootstrap in Classification Problems

Borislava Vrigazova
Sofia University, Faculty of Economics and Business Administration, Bulgaria

Abstract

Background: The bootstrap can be alternative to cross-validation as a training/test set splitting method since it minimizes the computing time in classification problems in comparison to the tenfold cross-validation. **Objectives:** This research investigates what proportion should be used to split the dataset into the training and the testing set so that the bootstrap might be competitive in terms of accuracy to other resampling methods. **Methods/Approach:** Different train/test split proportions are used with the following resampling methods: the bootstrap, the leave-one-out cross-validation, the tenfold cross-validation, and the random repeated train/test split to test their performance on several classification methods. The classification methods used include the logistic regression, the decision tree, and the k-nearest neighbours. **Results:** The findings suggest that using a different structure of the test set (e.g. 30/70, 20/80) can further optimize the performance of the bootstrap when applied to the logistic regression and the decision tree. For the k-nearest neighbour, the tenfold cross-validation with a 70/30 train/test splitting ratio is recommended. **Conclusions:** Depending on the characteristics and the preliminary transformations of the variables, the bootstrap can improve the accuracy of the classification problem.

Keywords: the bootstrap; classification; cross-validation; repeated train/test splitting

JEL classification: C38, C52, C55
Paper type: Research article

Received: 12 Aug 2020
Accepted: 15 Mar 2021

Citation: Vrigazova, B. (2021), "The Proportion for Splitting Data into Training and Test Set for the Bootstrap in Classification Problems", Business Systems Research, Vol. 12, No. 1, pp. 228-242.
DOI: https://doi.org/10.2478/bsrj-2021-0015

Acknowledgments: The author expresses her gratitude to prof. Ivan Ivanov for his recommendations. Some of the results presented in this paper were also presented at the ENTRENOVA 2020 conference. This research contains extended experiments and results. The author expresses her gratitude to the reviewers for their valuable comments.

Introduction

Long computational time is a problem that often occurs in big datasets. Slow computation can occur due to many reasons. On the one hand, computationally exhaustive methods like the mixed linear integer approach can be used with classification methods (Maldonado et. al., 2014). On the other hand, the input data may be used in their original version and the differences among their units can slow down the computation. A third reason can be the presence of too many independent variables. To avoid those problems and reduce computational time in classification, some authors suggest improved versions of existing computationally exhaustive methods for classification (Maldonado et. al., 2014), standardization of independent variables to unify the input variables (James et. al.), and variable selection (Velliangiri et. al., 2019). These approaches can reduce the time for splitting the dataset into training and test set to evaluate the performance of the classification model.

Some evidence suggests that computing time in machine learning algorithms for classification can also depend on the resampling method used for splitting the data into training and test set (James et. al., 2013). For instance, the leave-one-out cross-validation produces the training and test sets slower than the tenfold cross-validation, and the prediction time increases (James et. al., 2013). This paper shows that the tenfold bootstrap procedure introduced in (Vrigazova and Ivanov, 2020a) can decrease the overall time for prediction in classification problems. The paper compares the behaviour of the tenfold bootstrap to other resampling methods like the tenfold cross-validation (James et. al., 2013), the leave-one-out (LOO) cross-validation (James et. al., 2013), and the repeated random train/test split procedure available in Python (Pedregosa et. al., 2011). They are applied to several classification methods like the logistic regression, decision tree classifier, and the k-nearest neighbours. The aims of this paper are first to check if the tenfold bootstrap has the computational advantage as a training/test splitting method in classification methods. Similar research was conducted for the Support Vector Machines, so this paper can be considered as an extension of (Vrigazova and Ivanov, 2020b). Secondly, to propose train/test split proportion for the bootstrap procedure to reduce computational time and preserve high accuracy of the classification model.

The next section reviews existing academic literature, section 3 presents the methodology, and sections 4 and 5 comment on the results and discuss the advantages and disadvantages of the proposed methodology. Section 6 concludes.

Literature review

The bootstrap was first introduced in 1979 by Efron (Efron, 1979). It has wide applications in various fields. For example, it can be used for inferring the unknown distribution of data, thus allowing confidence intervals to be built. One thousand iterations of the bootstrap can make data's distribution closer to the Gaussian distribution. As a result, the bootstrap is widely used in Monte Carlo simulations MacKinnon (2002). The bootstrap is also used in the random forest classifier and for pruning decision trees (Breiman, 1996). In 1992, Breiman (1992) devised the little bootstrap procedure for applications as a resampling method in small datasets. Later, in 1995, he showed that the little bootstrap procedure can be used as a resampling method in data with fixed regressors (Breiman, 1995). He recommended cross-validation as a resampling technique in datasets with random regressors. In 2018 Vrigazova (2018) showed that the little bootstrap procedure (Breiman, 1992) can successfully be used for feature selection in panel data with fixed effects.

The bootstrap procedure has widely been used for estimating unknown distributions. Its properties as a resampling method have started to be more thoroughly researched lately. In 1997, Efron and Tibshirani (Efron et. al., 1997) tested the performance of the 0.632 + bootstrap procedure in machine learning methods for classification (k-nearest neighbour, logistic regression, and decision tree) suggesting that the bootstrap can be an alternative to cross-validation. Since then few experiments have been made in this direction. The standard resampling procedure for splitting the dataset into training and test set in classification problems has been cross-validation. Repeated random training/test split is also used as an alternative to cross-validation.

Based on the research of Efron and Tibshirani (Efron et. al., 1997), the question of the bootstrap procedure can be used as a technique for splitting into training and test set and be a reliable alternative to cross-validation has been raised. Recent research (Vrigazova and Ivanov, 2020a and b) has shown that the bootstrap procedure can be a reliable resampling procedure in the logistic regression, decision tree, k-nearest neighbour, and the support vector machines when using 70/30 proportion for train/test split. However, more experiments need to be conducted to conclude whether bootstrap is an appropriate training/test set splitting technique for various types of datasets. Also, it is subject to further experiments whether the 70/30 training/set proportion is appropriate in most cases to preserve high accuracy. This paper aims to fill these gaps in the academic literature.

Methodology

This research compares the performance of the decision tree classifier (James et. al., 2013), the logistic regression, and the k-nearest neighbour (Pampel, 2000) in terms of time, accuracy, and error rate. Logistic regression (Pampel, 2002) is a method for binary or multiclass classification based on the probability that one observation belongs to a particular class. It is relatively easier for interpretation than the decision tree classifier and the k-nearest neighbour. The decision tree classifier (James et.al., 2013) is not a computationally expensive method but it provides the predicted classes as a tree with possible outcomes leading to each class. Each branch of the tree is a particular variable. Therefore, it may be harder for interpretation particularly in the case of multiclass classification. Unlike the logistic regression and the decision tree classifier, the k-nearest neighbour splits the observations into classes based on how close they are to one another. It assumes that observations belonging to the same class will be close to one another. Typically, the three classification methods use tenfold cross-validation to split the dataset into training and test set and make predictions (James et. al., 2013). This paper investigates whether the tenfold bootstrap can be used instead of the tenfold cross-validation to split the dataset into training and test set so that the time for the prediction can be reduced.

To perform the research three fully available datasets were used. These are the Monica[1], the Food[2] , and the Adult[3] datasets. The Monica dataset is the smallest one, containing 6,367 observations and 11 independent variables. The dependent variable is called 'outcome". The Food dataset contains 23,971 observations and 5 independent variables, with the 'sex' variable being the dependent one. The last dataset is the Adult dataset with 45,222 observations and 11 independent variables. The dependent variable is 'income'. All datasets are increasing in size so that the

[1] Available at https://www.kaggle.com/ukveteran/who-monica-data/tasks

[2] Available at https://vincentarelbundock.github.io/Rdatasets/doc/Ecdat/BudgetFood.html

[3] Available at https://archive.ics.uci.edu/ml/datasets/adult

performance of the resampling methods in large datasets can be observed. The author did not apply preliminary transformations to the input variables.

All experiments were conducted in Python 3.6 using a computer with a processor Intel Core i7, 2.80 GHz., Windows 10. Time is measured in seconds, while accuracy and error rate are shown in equations 1 and 2.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{1}$$

$$Error\ rate = 1 - accuracy \tag{2}$$

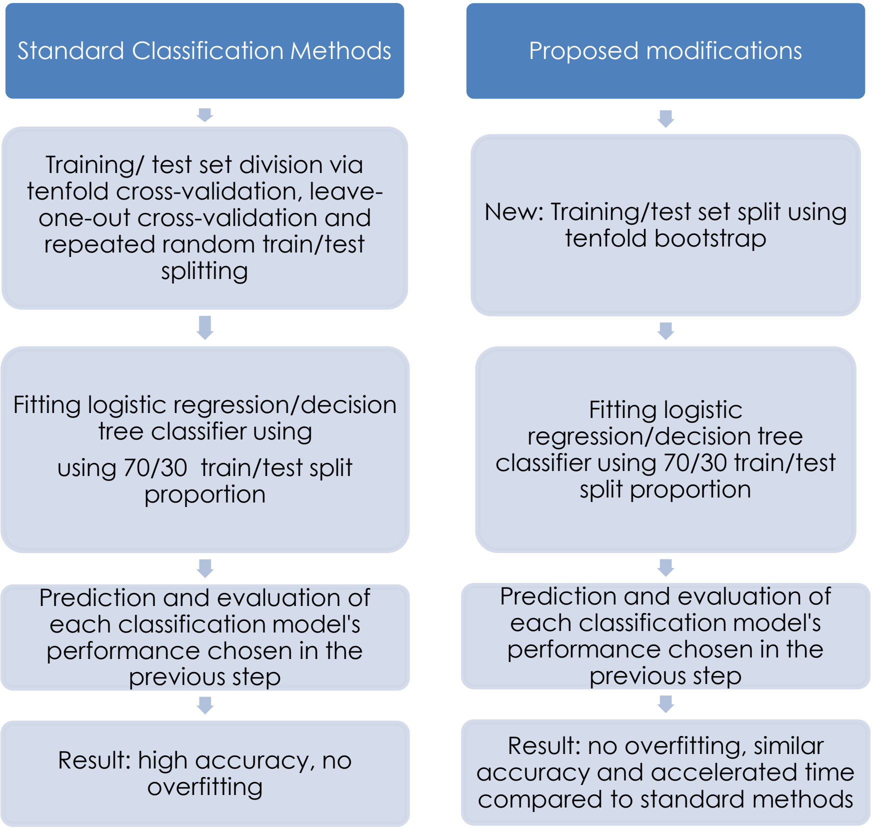
The first type of experiment is to split each dataset into training and test set using the tenfold cross-validation (Hoerl et. al., 1997) and 70/30, 50/50, 30/70, and 20/80 as train/test split proportions. The author then fitted each classification method and calculated time, accuracy, and error rate. The author used the Python 3.6 function model_selection.cross_val_score() with the parameter cv fixed to 10 to perform the tenfold cross-validation.

The leave-one-out (LOO) cross-validation was also used (Wong, 2015) as an alternative to tenfold cross-validation. The author used the same train/test split proportions as in the tenfold cross-validation. To run the leave-one-out (LOO) cross-validation, the function model_selection.LeavePOut(p=1) in Python was used with the parameter p set to 1. Then the leave-one-out cross-validation was applied to the three classification methods.

As a third resampling alternative, the repeated random train/test split (Krstajic et. al., 2015) was applied to the logistic regression, decision tree classifier, and the k-nearest neighbour. The function ShuffleSplit() can be used to randomly and repeatedly divide the dataset into training and test set. The author fixed the parameter n_splits to 10 and the random_state parameter to 7 to be able to replicate the results.

The author also ran the tenfold bootstrap (Vrigazova and Ivanov, 2019) procedure as an alternative to the three resampling methods. The bootstrap procedure for classification problems that this research follows was introduced in (Vrigazova and Ivanov, 2019). This paper shows that for some datasets the standard splitting proportion of 70/30 is not enough to optimize the performance of the bootstrap procedure. Other proportions may preserve accuracy, while further reduce computational time. Figure 1 summarizes the standard approach and the novel approach in this study.

Figure 1
Standard vs proposed resampling methods


Standard Classification Methods
Training/ test set division via tenfold cross-validation, leave-one-out cross-validation and repeated random train/test splitting
Fitting logistic regression/decision tree classifier using
using 70/30 train/test split proportion
Prediction and evaluation of each classification model's performance chosen in the previous step
Result: high accuracy, no overfitting
Proposed modifications
New: Training/test set split using tenfold bootstrap
Fitting logistic regression/decision tree classifier using 70/30 train/test split proportion
Prediction and evaluation of each classification model's performance chosen in the previous step
Result: no overfitting, similar accuracy and accelerated time compared to standard methods

Source: Author's presentation

To compare the performance of each model, the author uses time, accuracy, and error rate. The next section presents the results.

Results

Logistic regression

Table 1 presents the results from the resampling methods applied to the logistic regression.

Table 1 shows that the slowest resampling method is the leave-one-out cross-validation (LOO). Regardless of the size of the dataset and the splitting proportion, the leave-one-out cross-validation was between 18 and 6440 times slower than the rest of the resampling methods. Despite this, it produced an accuracy and error rate similar to the tenfold cross-validation. Its computational disadvantage makes it rarely used in large datasets. The tenfold cross-validation is faster than the leave-one-out cross-validation but slower than the random train/test split and the tenfold bootstrap.

The tenfold bootstrap proved to be the fastest resampling method for the logistic regression. Its computational advantage was significant. For instance, the adult dataset (70/30) was classified by the LOO in 6440 seconds, while the bootstrap did that in 0.23 seconds. The tenfold cross-validation led to the output from the logistic regression in 1.78 seconds, while the random train/test split produced results similar to the bootstrap. The two produced an accuracy of 79.1%, while the cross-validation – 79.8%. However, the accuracy of the bootstrap is stable regardless of the splitting proportion, similarly to the random train/test split. Unlike them, the tenfold cross-validation's accuracy fell from 79.8% to 79.2%. So, possible overfitting can be present in the cross-validation.

Table 1
Logistic regression results

Dataset	Train/test ratio	Resampling method	Accuracy	Error rate	Time (s)
Monica	70/30	10-fold cross-validation	87.8	12.2	1.84
		LOO	87.8	12.2	105.56
		Random train/test split	87.9	12.1	0.05
		10-fold bootstrap	87.8	12.2	0.02
	50/50	10-fold cross-validation	87.7	12.3	0.14
		LOO	87.7	12.3	44.70
		Random train/test split	87.9	12.1	0.05
		10-fold bootstrap	87.4	12.6	0.01
	30/70	10-fold cross-validation	87.9	12.1	0.09
		LOO	87.9	12.1	18.32
		Random train/test split	88.0	12.0	0.14
		10-fold bootstrap	87.5	12.5	0.01
	20/80	10-fold cross-validation	87.8	12.2	0.05
		LOO	88.0	12.0	7.68
		Random train/test split	87.4	12.6	0.04
		10-fold bootstrap	87.5	12.5	0.01
Food	70/30	10-fold cross-validation	86.2	13.8	0.83
		LOO	86.2	13.8	306.52
		Random train/test split	86.4	13.6	0.05
		10-fold bootstrap	86.1	13.9	0.03
	50/50	10-fold cross-validation	86.2	13.8	0.10
		LOO	86.2	13.8	145.48
		Random train/test split	85.8	14.2	0.15
		10-fold bootstrap	86.1	13.9	0.02
	30/70	10-fold cross-validation	86.3	13.7	0.07
		LOO	86.3	13.7	55.77
		Random train/test split	86.0	14.0	0.04
		10-fold bootstrap	86.0	14.0	0.01
	20/80	10-fold cross-validation	86.1	13.9	0.06
		LOO	86.1	13.9	28.24
		Random train/test split	86.0	14.0	0.04
		10-fold bootstrap	86.0	14.0	0.01
Adult	70/30	10-fold cross-validation	79.8	20.2	1.78
		LOO	79.7	20.3	6440.27
		Random train/test split	79.1	20.9	0.23
		10-fold bootstrap	79.1	20.9	0.23
	50/50	10-fold cross-validation	79.7	20.3	0.99
		LOO	79.7	20.3	3029.14
		Random train/test split	79.0	21.0	0.19
		10-fold bootstrap	79.2	20.8	0.12
	30/70	10-fold cross-validation	79.5	20.5	0.41
		LOO	79.6	20.4	659.80
		Random train/test split	79.1	20.9	0.14
		10-fold bootstrap	79.2	20.8	0.07
	20/80	10-fold cross-validation	79.2	20.8	0.30
		LOO	79.2	20.8	273.80
		Random train/test split	79.1	20.9	0.10
		10-fold bootstrap	79.3	20.7	0.06

Source: Author's calculations

The accuracy did not change so drastically with reducing the training set. All resampling methods provided an error rate between 13.6% and 14%. The bootstrap resulted in the highest accuracy of 86.1% (70/30), while the tenfold cross-validation – 86.2% (70/30). The random train/test split resulted in an accuracy of 86.4% (70/30). However, when the train/test random split was applied with a 50/50 splitting proportion, its accuracy dropped to 85.8%. The 30/70 proportion led to increased accuracy (86.3%) resulting from the tenfold cross-validation. Changing the splitting proportion did not lead to significant changes in the logistic regression's error rate but significantly accelerated the computing time. It accelerated the logistic regression to be 306 times faster than the leave-one-out cross-validation and 27 times faster than the tenfold cross-validation.

Splitting the dataset into 70/30 proportion led to 87.8% accuracy from the cross-validation and the bootstrap. The exception was the leave-one-out cross-validation that produced an accuracy of 87.9%. When using a smaller training set, the random train/test split resulted in 88% accuracy, while the other methods had a slight increase. However, the bootstrap procedure was the fastest. Using splitting proportions of 50/50, 30/80, and 20/80 did not cause the bootstrap to reduce accuracy significantly. However, computational time decreased compared to the 70/30 proportion. The bootstrap procedure in the logistic regression had relatively stable performance in terms of accuracy regardless of the train/test split proportion. However, other resampling methods lost accuracy as the training set decreased.

The author considers the bootstrap procedure as suitable for train/test set split for the logistic regression in a large dataset as it provided similar results to the tenfold cross-validation that did not change much with the decreasing of the size of the training set. Therefore, the author recommends using the 30/70, 20/80, and 50/50 proportions to preserve accuracy, while further decreasing computational time.

The Decision Tree Classifier

Similar observations can be made for the decision tree classifier. Table 2 summarizes its performance.

The bootstrap optimizes the performance of the decision tree classifier as well. The bootstrap produced the output from the decision tree classifier (70/30) in 0.17 seconds on the adult dataset, while the tenfold cross-validation in 1.65 seconds. As table 2 shows the bootstrap resulted in an accuracy and error rate, similar to those from the other resampling methods. However, the computational time was much faster. In some cases, the bootstrap decreased the error rate of the model.

Like the logistic regression, the accuracy of the decision tree classifier started to increase as a result of the cross-validation and the random train/test split when the size of the training set decreased. With the decrease of the size of the training set, cross-validation and the random train/test set tended to overfit, which increased the model's accuracy. However, the performance of the bootstrap remained relatively unchanged with the decrease of the training set size and close to the accuracy from the standard cross-validated 70/30 version of the decision tree.

The author believes the reason behind this result is that the bootstrap can reduce overfitting even when the training set is smaller than the test set. It is important to be noted that the datasets did not have any preliminary transformations. In previous research, Vrigazova and Ivanov (2020a) showed that if the input data have been standardized and variable selection is performed, the bootstrap produces higher accuracy than other resampling methods.

Table 2
Resampling methods for the Decision Tree Classifier

Dataset	Train/test ratio	Resampling method	Accuracy	Error rate	Time
Monica	70/30	10-fold cross-validation	80.7	19.3	0.08
		LOO	80.8	19.2	40.92
		Random train/test split	81.3	18.7	0.03
		10-fold bootstrap	80.5	19.5	0.01
	50/50	10-fold cross-validation	80.9	19.1	0.07
		LOO	81.7	18.3	20.40
		Random train/test split	80.5	19.5	0.03
		10-fold bootstrap	80.6	19.4	0.01
	30/70	10-fold cross-validation	81.5	18.5	0.05
		LOO	82.1	17.9	8.11
		Random train/test split	80.5	19.5	0.03
		10-fold bootstrap	80.5	19.5	0.01
	20/80	10-fold cross-validation	81.2	18.8	4.20
		LOO	80.1	19.9	0.02
		Random train/test split	80.0	20.0	0.01
		10-fold bootstrap	80.5	19.5	0.01
Food	70/30	10-fold cross-validation	83.5	16.5	0.67
		LOO	83.6	16.4	1383.83
		Random train/test split	83.9	16.1	0.14
		10-fold bootstrap	83.7	16.3	0.09
	50/50	10-fold cross-validation	83.5	16.5	0.48
		LOO	83.5	16.5	635.69
		Random train/test split	83.9	16.1	0.10
		10-fold bootstrap	83.7	16.3	0.06
	30/70	10-fold cross-validation	83.7	16.3	0.26
		LOO	83.2	16.8	211.96
		Random train/test split	83.7	16.3	0.07
		10-fold bootstrap	83.5	16.5	0.04
	20/80	10-fold cross-validation	83.7	16.3	0.17
		LOO	83.6	16.4	100.17
		Random train/test split	83.8	16.2	0.05
		10-fold bootstrap	83.5	16.5	0.03
Adult	70/30	10-fold cross-validation	80.9	19.1	1.65
		LOO	80.6	19.4	4815.19
		Random train/test split	79.6	20.4	0.25
		10-fold bootstrap	80.4	19.6	0.17
	50/50	10-fold cross-validation	80.5	19.5	0.86
		LOO	80.5	19.5	2566.74
		Random train/test split	79.8	20.2	0.19
		10-fold bootstrap	80.4	19.6	0.12
	30/70	10-fold cross-validation	80.8	19.2	0.49
		LOO	80.7	19.3	858.83
		Random train/test split	79.6	20.4	0.12
		10-fold bootstrap	80.2	19.8	0.08
	20/80	10-fold cross-validation	79.8	20.2	0.33
		LOO	79.8	20.2	339.88
		Random train/test split	79.0	21.0	0.10
		10-fold bootstrap	80.0	20.0	0.06

Source: Author's calculations

The bootstrap produced similar accuracy to that of the tenfold cross-validation regardless of the splitting proportion. The cross-validation methods and the random train/test split varied in accuracy depending on the splitting ratio. Therefore, the bootstrap can also be applied with other splitting proportions like the ones presented in this research. The bootstrap procedure can avoid overfitting not only by using a smaller training set but also by using nontransformed data as tables 1 and 2 suggest.

Some research (Vrigazova and Ivanov, 2020a) suggests that the bootstrap applied with a 30/70 splitting proportion can also preserve accuracy while decreasing computing time. The authors there show that the support vector machines classifier with tenfold bootstrap and 30/70 splitting ratio can produce similar accuracy to that produced from the tenfold cross-validation with a ratio of 70/30. The advantage is the computing time. As tables 1 and 2 show, this paper confirmed this finding for the logistic regression and the decision tree classifier as well. However, when applied to untransformed data without variable selection to the logistic regression and the decision tree classifier, the bootstrap can be used with a 50/50 splitting ratio instead of 30/70. Depending on the characteristics of the dataset, other proportions can also be suitable as tables 1 and 2 show.

This is an important finding as the bootstrap can additionally decrease computing time by applying a smaller size of the training set but preserve the accuracy of the model. The other resampling methods suffer from fluctuations, so changing the splitting ratio affects the error rate and may cause overfitting. As the tables show the computing time decreased but the accuracy either fell, either increased. The bootstrapped classification is affected by non-transformed data the least while reducing further computational time.

The K-nearest Neighbour

Table 3 presents the results for the k-nearest neighbour. As the table shows, the bootstrap procedure used with a 70/30 splitting proportion was faster than the tenfold cross-validation with a 70/30 split. However, the bootstrap's performance in the k-nearest neighbour was not so good compared to the logistic regression and the decision tree. The bootstrap with a 70/30 split proportion resulted in about 2 percentage points higher error rate than the tenfold cross-validation. This finding is in line with (Vrigazova and Ivanov, 2020a).

However, increasing the size of the test set did not lead to significant improvement of the accuracy from the bootstrap. The leave-one-out cross-validation and the repeated training/test split produced better accuracy than the bootstrap. Despite this, the bootstrap procedure was the fastest. This result is not surprising as previous research (Vrigazova and Ivanov, 2020a) suggested that the bootstrap procedure may not be suitable for the k-nearest neighbour as a resampling method using the 70/30 train/test split proportion. We extended the research of these authors by confirming, firstly, that changing the training/test split proportion cannot increase the accuracy of the bootstrapped k-nearest neighbour.

Secondly, the bootstrap may not be a suitable resampling method for the k-nearest neighbour. Our experiments suggest that the most suitable resampling method for the k-nearest neighbour is the tenfold cross-validation with a train/test splitting proportion of 70/30. Although the tenfold cross-validation was slower than the bootstrap as table 3 shows, it resulted in high accuracy and was relatively faster than the leave-one-out cross-validation. Although the repeated train/test split was faster than the tenfold cross-validation, it produced lower accuracy. Thus, we recommend using the tenfold cross-validation with 70/30 splitting ratio as a resampling method in the k-nearest neighbours.

Table 3
Resampling methods for the K-nearest Neighbour

Dataset	Train/test ratio	Resampling method	Accuracy	Error rate	Time
Monica	70/30	10-fold cross-validation	79.9	20.1	0.44
		LOO	80.1	19.9	26.26
		Random train/test split	80.0	20.0	0.21
		10-fold bootstrap	77.5	22.5	0.05
	50/50	10-fold cross-validation	78.6	21.4	0.07
		LOO	79.3	20.7	13.41
		Random train/test split	78.4	21.6	0.26
		10-fold bootstrap	76.6	23.4	0.04
	30/70	10-fold cross-validation	78.7	21.3	0.05
		LOO	80.1	19.9	26.07
		Random train/test split	76.8	23.2	0.26
		10-fold bootstrap	76.3	23.7	0.04
	20/80	10-fold cross-validation	76.8	23.2	0.04
		LOO	80.8	19.2	33.05
		Random train/test split	74.8	25.2	0.26
		10-fold bootstrap	74.5	25.5	0.03
Food	70/30	10-fold cross-validation	85.1	14.9	0.32
		LOO	84.9	15.1	486.95
		Random train/test split	84.8	15.2	0.08
		10-fold bootstrap	83.0	17.0	0.06
	50/50	10-fold cross-validation	84.9	15.1	0.21
		LOO	84.9	15.1	201.47
		Random train/test split	84.7	15.3	0.10
		10-fold bootstrap	83.6	16.4	0.06
	30/70	10-fold cross-validation	85.1	14.9	0.12
		LOO	84.8	15.2	62.31
		Random train/test split	85.0	15.0	0.10
		10-fold bootstrap	84.0	16.0	0.07
	20/80	10-fold cross-validation	84.5	15.5	0.08
		LOO	85.0	15.0	28.56
		Random train/test split	85.1	14.9	0.12
		10-fold bootstrap	84.4	15.6	0.06
Adult	70/30	10-fold cross-validation	77.4	22.6	2.06
		LOO	77.3	22.7	6373.29
		Random train/test split	75.7	24.3	0.36
		10-fold bootstrap	74.4	25.6	0.46
	50/50	10-fold cross-validation	77.4	22.6	1.31
		LOO	77.3	22.7	2624.45
		Random train/test split	75.6	24.4	0.41
		10-fold bootstrap	74.8	25.2	0.45
	30/70	10-fold cross-validation	76.2	23.8	0.47
		LOO	76.3	23.7	572.23
		Random train/test split	75.1	24.9	0.45
		10-fold bootstrap	74.5	25.5	0.49
	20/80	10-fold cross-validation	76.0	24.0	0.26
		LOO	75.9	24.1	196.10
		Random train/test split	74.8	25.2	0.45
		10-fold bootstrap	75.1	24.9	0.40

Source: Author's calculations

Discussion

This paper proposes a new approach to accelerate computational time in classification models. The issue of slow computational time becomes severe in large datasets, where classification can take days and months. Existing literature uses various approaches to solve this issue. Most of them include changing the equation of the classification model, using a different type of model, dimensionality reduction, or data transformation.

For instance, the mixed linear integer approach (Iannarilli & Rubin, 2003) is a mathematical method that is known to be computationally exhaustive. However, mathematical methods can be modified to be applied in combination with machine learning classification, while reducing computational time. The mixed linear integer approach in classification models (Iannarilli & Rubin, 2003) was modified by Maldonado (2014) so that it might be used in classification models but perform faster prediction than the version of Iannarilli & Rubin (2003).

Despite the adaptation of the mixed linear integer approach to classification models, it still performs slower predictions than traditional machine learning methods (Vrigazova & Ivanov, 2020b). Therefore, improving one class of methods does not guarantee the fastest classification. Another approach for reducing computational time in classification problems that academic literature recommends is changing the type of model. For example, both the logistic regression and the decision tree classifier can be appropriate for a particular dataset but the decision tree classifiers can be faster as they are not a computationally expensive method (Grubinger et. al., 2014). The logistic regression, however, can be interpreted more easily. Depending on the aim of the research, the researcher needs to decide whether he/she will use a computationally inexpensive method.

Another approach to reducing computational time in classification is by using dimensionality reduction techniques. They can be built-in in the classification model (Kim & Shin, 2019) or used as a preprocessing step (Yeturu, 2020). Dimensionality reduction techniques may include feature selection, feature ranking, and principal component analysis (Yeturu, 2020). These methods aim to choose the features that carry the most important information for the prediction. Therefore, a subset of the independent variables is produced that is later used in classification. With the reduction of features, the classification model becomes less computationally expensive (Yeturu, 2020). However, the focus of this approach is not to reduce computational time but rather to improve the classification metrics like accuracy.

Preliminary transformations of data like standardization can also reduce computational time by limiting high fluctuations in data and transforming the features to have small values that do not require computationally expensive calculations (James et. al., 2013). However, this approach is not widely used for reducing computational time as it has become a standard step in the building of a classification model (Yeturu, 2020). As Wong (2015) and machine learning textbooks (Yeturu, 2020), (James et. al., 2013) stated, the velocity for making a prediction depends also on the resampling procedure used for splitting the dataset into training and test set. For instance, the leave-one-out cross-validation (Wong, 2015), (James et. al., 2013), (Yenturu, 2020) is computationally expensive, which slows down predictions. This result is confirmed in this research (tables 1-3). Their work raises the question of whether another resampling method can reduce computational time without loss of accuracy. This paper provided an answer to this question.

The results in this paper extend existing academic literature (Wong, 2015), (James et. al., 2013), (Yenturu, 2020) by proposing a new practical application of bootstrap as a training/test splitting method that reduces computational time in classification.

The paper shows that changing the resampling method can be another approach to solve the issue with long computation in classification problems. This result has important implications in a large dataset as the bootstrap can lead to a much faster result than the cross-validation and the repeated random train/test split. The paper shows that the random repeated train/test split method is faster than the tenfold cross-validation and the leave-one-out cross-validation but slower than the bootstrap. The random repeated train/test split algorithm leads to loss of accuracy in some cases, while the bootstrap resulted in similar accuracy to that from the tenfold cross-validation. Another important recommendation from this paper is using the bootstrap as a resampling method with a 70/30 train/test split proportion to achieve the best results. To the author's best knowledge, this research has been the most detailed one concerning the applications of bootstrap in machine learning classification. A very important finding from the research is that the bootstrap is suitable for the logistic regression and the decision tree classifier but it causes loss of accuracy in the k-nearest neighbours. With this, the paper recognizes not only the advantages of the bootstrap in classification problems but for the first time, it outlines a case, where it may not be suitable for use.

Several limitations of the approach in this paper should be noted, however. The first one is that input data were not transformed. When standardized, for example, the accuracy resulting from the four resampling methods may change. The bootstrap may result in better accuracy than that achieved by the rest of the resampling method. Although the author has a reason to believe that can be the case, this hypothesis should be checked. Therefore, a further direction of this research would be to check what happens with time and accuracy when data are standardized. Second, standardization of data combined with the bootstrap can also affect the outcome from the k-nearest neighbour. Standardized data and the bootstrap may preserve or increase the accuracy of the k-nearest neighbour. This hypothesis should also be checked.

Also, this paper proposes the use of ten iterations of the bootstrap. It should be noted that ten iterations are enough to preserve the accuracy of the model. Increasing the number of iterations can result in computationally exhaustive classification. For example, running 100 iterations of the bootstrap may result in similar or slightly better average accuracy than that from the tenfold bootstrap but the time would increase. The author chose ten iterations of the bootstrap to be comparable to the tenfold cross-validation in terms of the number of iterations.

It is also possible that the proposed approach may not be suitable for some datasets despite using the logistic regression and the decision tree classifier. A future extension of this research would be to examine how the bootstrap would affect the outcome of the decision tree when it is pruned. Also, are those findings valid in the case of multiclass classification? The paper proposes using 70/30 proportion to split the dataset into training and test set. However, depending on the characteristics of the data and their preliminary transformations, this proportion can differ from dataset to dataset. This is hardly a limitation of this paper as machine learning textbooks (James et. al., 2013) do not provide a rule for selecting the training/test splitting ratio. Therefore, despite providing good results on the datasets used in this research, the 70/30 ratio may not be suitable for all kinds of datasets.

Despite the limitations of this research, it has very important practical implications. As tables, 1-3 show the bootstrap can reduce computational time several times compared to the cross-validation and the repeated random train/test split while preserving accuracy high. This finding is important as the tenfold bootstrap can perform classification in large datasets without variable selection much faster than the tenfold and leave-one-out cross-validation. This allows the proposed methodology to be used either as a way to quickly acquaint with the data, for model specification (e.g. the logistic regression/ decision tree classifier) or as a predictive model with reduced computing time. All these advantages of the bootstrap procedure allow it to be a powerful tool in performing machine learning classification models.

Conclusion

In this paper, it is shown that using a smaller training set with the bootstrap can preserve high accuracy and further decrease the computational time of the classification model. The advantages of using 50/50, 30/70, and 20/80 ratio as training/test set splitting proportions with the bootstrap procedure, however, are valid only for the logistic regression and the decision tree classifier. Using the bootstrap procedure as a resampling method in the k-nearest neighbour is not recommended due to loss of accuracy. Instead, this research recommends that the k-nearest neighbour might be fitted by using the tenfold cross-validation with a train/test splitting ratio of 70/30.

Using a 20/80 training/test ratio differs from academic literature and machine learning textbooks as the number of training instances have to be large enough to make correct predictions of the test data. A small number of training observations may fail to capture all important characteristics of the data and make incorrect predictions. However, the bootstrap procedure allows for correct predictions even when the training set contains 20% of the dataset (in the case of the logistic regression and the decision tree classifier). Also, the academic literature suggests improvements of existing versions of the logistic regression and the decision tree classifier to reduce computational time but they usually do not involve a change of the resampling method.

The k-fold cross-validation has become the standard resampling method used in both the classic versions of the logistic regression and the decision tree classifier and their modifications. The reason for this is that the k-fold cross-validation provides a reasonable balance between accuracy and computational time. The experiments conducted in this research show that the tenfold bootstrap has similar advantages in the case of the logistic regression and the decision tree classifier. On one hand, the bootstrap resulted in similar accuracy as the tenfold cross-validation, while performing faster classification than other resampling methods, including the tenfold cross-validation. This advantage of the bootstrap can be observed using various training/test split proportions, e.g. 20/80. These findings have important practical implications in large datasets as the bootstrap complements existing academic literature by extending the ways for accelerating fitting and making predictions with the logistic regression and the decision tree classifier.

Despite the practical advantages of the tenfold bootstrap as a resampling method, several disadvantages should be considered. Depending on the characteristics of the dataset, the 20/80 splitting proportion may not always guarantee high accuracy. So, the rule for a larger training set than the test set may be valid using the tenfold bootstrap as well. The best training/test set splitting proportion via the bootstrap can differ in each dataset. Also, preliminary transformations of data may affect the accuracy of the model. Thus, it is possible that if independent variables are standardized, the accuracy of the classification may be increased even in the case of the k-nearest neighbours. A further step of this research would be to check if standardization of data can increase the accuracy of the bootstrap procedure. If so, the advantages of the tenfold bootstrap as a resampling method in classification problems can be further extended.

References

1. Breiman L., (1995), "Better Subset Regression Using the Nonnegative Garrote", Technometrics, Vol. 37 No 4, pp. 373 – 384.
2. Breiman L., (1992), "The Little Bootstrap and Other Methods for Dimensionality Selection in Regression: X-fixed Prediction Error", Journal of American Statistical Association, Vol. 87 No. 419, pp. 738 -754.
3. Breiman, L. (1996), "Bagging predictors", Machine Learning. 24 (2), pp. 123–140.
4. Grubinger, T., Zeileis, A. and Pfeiffer, K., 2014. Evtree: Evolutionary learning of globally optimal classification and regression trees in R. J. Stat. Software 61 (1), pp. 1-29.
5. Efron B., (1979), "Bootstrap Methods: Another Look at the Jackknife", the Annals of Statistics, Vol. 17, pp. 1–26.
6. Efron B., Tibshirani R., (1997), "Improvements on Cross-Validation: The .632+ Bootstrap Method", Journal of the American Statistical Association, vol. 92, pp. 548–560.
7. Hoerl E., Kennard W., (1970), "Ridge Regression. Applications to nonorthogonal Problems", Technometrics, Vol. 12 No. 1, pp. 69-82. Iz 2012
8. Iannarilli F., Rubin P., (2003), Feature selection for multiclass discrimination via mixed-integer linear programming, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25 No. 6, pp. 779-783.
9. James G., D. W., Hastie T., Tibshirani R., (2013), An Introduction to Statistical Learning. Springer, STS Vol. 103.
10. Kim B., Shin S., (2019), "Principal weighted logistic regression for sufficient dimension reduction in binary classification", Journal of the Korean Statistical Society, Vol. 48 No. 2, pp. 194-206.
11. Krstajic D., Buturovic J., Leahy E., Thomas S., (2014), "Cross-validation pitfalls when selecting and assessing regression and classification models", Cheminformatics, Vol. 6 Article No. 10.
12. MacKinnon J., (2002), "Bootstrap Inference in Econometrics", The Canadian Journal of Economics, Vol. 35 No. 4, pp. 615—645.
13. Maldonado S., Pérez J., Weber R., Labbé M., (2014), Feature Selection for Support Vector Machines via Mixed Integer Linear Programming, Information Sciences, Vol. 279, pp. 163–175.
14. Pampel F., (2000), Logistic regression: A primer. Sage University Papers Series on Quantitative Applications in the Social Sciences, 07-132. Sage Publications, Thousand Oaks, CA.
15. Pedregosa et al., (2011), Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12, pp. 2825-2830.
16. Velliangiri S., Alagumuthukrishnan S., Joseph S., (2019), A Review of Dimensionality Reduction Techniques for Efficient Computation, Procedia Computer Science, Vol. 165, pp. 104-111.
17. Vrigazova B., (2018), "Nonnegative Garrote as a Variable Selection Method in Panel Data", International Journal of Computer Science and Information Security, Vol. 16 No. 1.
18. Vrigazova B., Ivanov I., (2019), "Optimization of the ANOVA Procedure for Support Vector Machines", International Journal of Recent Technology and Engineering, Vol. 8 No. 4.
19. Vrigazova B., Ivanov I., (2020a), "The bootstrap procedure in classification problems", International Journal of Data Mining, Modelling and Management, Vol. 12 No. 4.
20. Vrigazova, B.& Ivanov, I., (2020b), "Tenfold bootstrap procedure for support vector machines", Computer Science, Vo. 21 No. 2, pp. 241-257. 10.7494/csci.2020.21.2.3634.
21. Wong T., (2015), "Performance evaluation of classification algorithms by k-fold and leave-one-out cross-validation", Pattern Recognition, Vol. 48 No. 9, pp. 2839–2846.
22. Yeturu K., (2020), Chapter 3 - Machine learning algorithms, applications, and practices in data science, Editor(s): Arni S.R. Srinivasa Rao, C.R. Rao, Handbook of Statistics, Elsevier, Vol. 43, pp. 81-206.

About the author

Borislava Vrgazova is a data science practitioner. Her research areas include practical applications of machine learning algorithms for prediction and how their performance can be boosted. Also, applications of big data techniques to small datasets in the field of economics as an alternative to traditional econometrics theory. She challenges traditional econometric modelling techniques used to find connections among variables from institutional economics by combining feature selection methods and big data prediction models. As a result, new applications of machine learning techniques to economic data appear. The author can be contacted at **vrigazova@uni-sofia.bg**