DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2020

A Comparative study of data splitting algorithms for machine learning model selection

DELWENDE ELIANE BIRBA

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Abstract

Data splitting is commonly used in machine learning to split data into a train, test, or validation set. This approach allows us to find the model hyper-parameter and also estimate the generalization performance. In this research, we conducted a comparative analysis of different data partitioning algorithms on both real and simulated data. Our main objective was to address the question of how the choice of data splitting algorithm can improve the estimation of the generalization performance. Data splitting algorithms used in this study were variants of k-fold, Kennard-Stone, SPXY (sample set partitioning based on joint x-y distance), and random sampling algorithm. Each algorithm was used to divide the data into two different sets, training, and validation sets. We then analyzed the different data splitting algorithms based on the generalization performances estimated from the validation and the external test set.

From the result, we noted that the important determinant for a good generalization is the size of the dataset. For all the data sample methods applied on small data set, the gap between the performance estimated on the validation and test set was significant. However, we noted that the gap reduced when there was more data in training or validation. Too many or few data in the training set can also lead to bad model performance. This highlights the importance to have a proper balance between the training/validation set sizes. In our study, KS and SPXY was the splitting algorithm with poor model performance estimation. Indeed these methods select the most representative samples to train the model, and poor representative samples are left for model performance estimation.

Keywords: K-fold, cross-validation, Kennard-Stone algorithm, data splitting, bootstrap, overfitting

Abstrakt

Datadelning används vanligtvis i maskininlärning för att dela data i en tåg-, test- eller valideringsuppsättning. Detta tillvägagångssätt gör det möjligt för oss att hitta hyperparametern för modellen och även uppskatta generaliseringsprestanda. I denna forskning genomförde vi en jämförande analys av olika datapartitionsalgoritmer på både verkliga och simulerade data. Vårt huvudmål var att ta upp frågan om hur valet av datasplitningsalgoritm kan förbättra uppskattningen av generaliseringsprestanda. Datadelningsalgoritmer som användes i denna studie var varianter av k-fald, Kennard-Stone (KS), SPXY (provuppsättning partitionering baserat på gemensamt x-y-avstånd) och bootstrap-algoritm. Varje algoritm användes för att dela upp data i två olika uppsättningar, tränings- och valideringsuppsättningar. Vi analyserade sedan de olika datasplittringsalgoritmerna baserat på generaliseringsprestanda uppskattade från valideringen och den externa testuppsättningen.

Från resultatet noterade vi att den viktiga determinanten för en bra generalisering är storleken på data. För alla dataprövmetoder som använts på små datamängder var klyftan mellan prestanda uppskattad på valideringen och testuppsättningen betydande. Vi noterade emellertid att gapet minskade när det fanns mer data i utbildning eller validering. För många eller få data i träningsuppsättningen kan också leda till dålig modellprestanda. Detta belyser vikten av att ha en korrekt balans mellan utbildnings- valideringsuppsättningsstorlekarna. I vår studie var KS och SPXY den delande algoritmen med dålig modellprestanda. Dessa metoder väljer faktiskt de mest representativa proverna för att träna modellen, och dåliga representativa prover lämnas för uppskattning av modellprestanda.

Nyckelord: K-fald, korsvalidering, Kennard-Stone-algoritm, datasplitning, bootstrap, överanpassning

Acknowledgements

This enterprise would not have been possible without the blessing of the Almighty to whom I owe all my success. I would like to express my deepest gratitude to my examiner, for his valuable advice and encouragement throughout this study. His exquisite knowledge and experience helped me to understand various critical issues related to research. I feel grateful to my supervisor, who helped me to chart a smooth path for my research.

Authors

Delwende Eliane Birba, delianeb1@gmail.com
Information and Communication Technology
KTH Royal Institute of Technology

Place for Project

Nice, France
ChemoSim lab

Examiner

Henrik Boström
KTH Royal Institute of Technology

Supervisor

KTH Supervisor: Erik Fransén
Intitute of chemistry of Nice: Jeremie Topin and Jerome Golebiowski

Contents

1 Introduction

This chapter presents a general background and problem formulation. Also, the organization of the thesis work is presented as well as objectives and purpose.

1.1 Background

Machine learning algorithms aim to extract knowledge from data and produce viable prediction models. One of the main goal is to build computational models with high prediction and generalization capabilities [18]. The generalization ability depends on the model complexity. The complexity of a machine learning model depends on his hyper-parameters. A model with high complexity can have a risk of over-fitting. However, data splitting into training and validation sets can help to find the most efficient set of model parameter(s), which has a correct balance between the model generalization capabilities and his complexity. The training set is used to build the model [10]. This data is used to fit the parameters of the model [16]. The model can see this data and learn from it. On the other hand, the validation set used to challenge the model. ”The validation set provides an unbiased evaluation of a model fit on the training dataset while tuning the model’s hyperparameters” [3]. This is the model selection procedure [19].

Recent research has shown that the validation set is not always enough to measure the model’s performance. Westerhuis et al. [20] have demonstrated that cross-validation can give an over-optimistic result. The study of Harrington et al. [9] proved that having only training and validation sets could also give a wrong estimation of model performance. These studies highlight the need to have another set. This external test set allows us to evaluate the generalization performance of the model on unseen data. Machine learning model validation process can be illustrated with flowchart in figure 1.1 [21]. However, even with the process described in (Fig. 1), it is still challenging to have an external data set with the same distribution as data in real-world applications. Indeed, the data structure, in real-world applications, is most of the time unknown in advance. Also, many other factors can affect the model generalization performance. The size of the different data sets, training validation, test set, and the data splitting algorithm can impact the model.

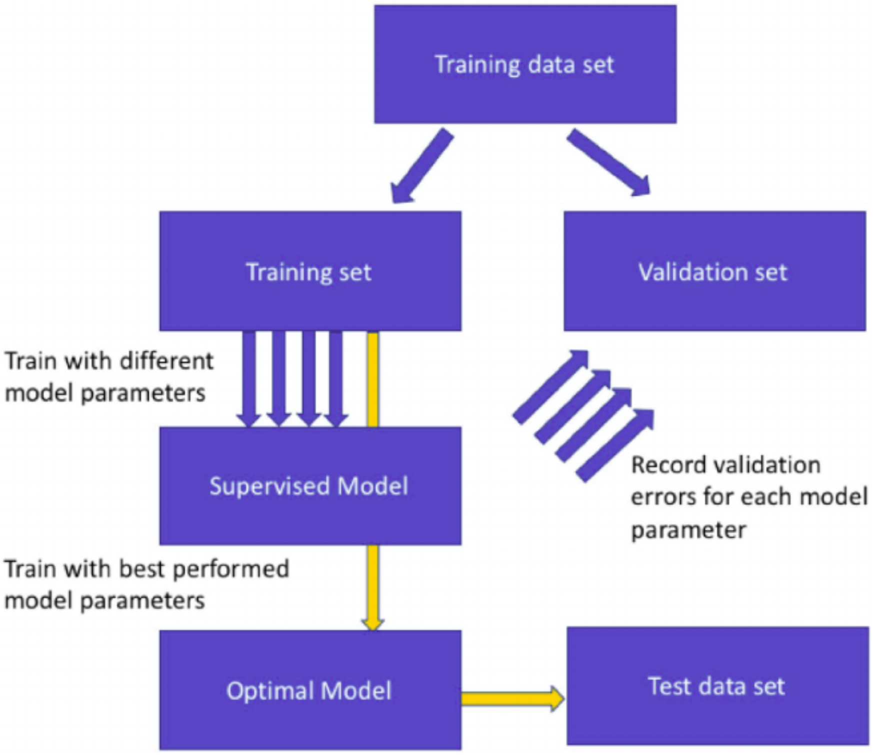
Training data set
Training set
Validation set
Train with different model parameters
Record validation errors for each model parameter
Supervised Model
Train with best performed model parameters
Optimal Model
Test data set

Figure 1.1: A model selection general flowchart. " arrows in blue color is the validation flow and yellow arrows the final training and evaluation with external test flow " source: [21].

1.2 Problem

Our study centered on three categories of data splitting algorithms: random splitting, cross-validation, and rational splitting algorithm. Random selection algorithms randomly select a set of samples a validation, and the remaining ones are used to fit the model. This process can be repeated many times. The selection can be with or without replacement. The bootstrap algorithm proposed by Efron et al. [8] is an example of a random splitting algorithm for our analysis. On the other hand, rational splitting algorithms select samples base on the similarity between the data point or the data distribution. The objective is to select the most representative sample for the training set. Kennard-Stone algorithm was used in our study as an example rational splitting algorithm. Cross-validation algorithm is described in the next chapter.

Most of the data splitting algorithms have a parameter that must be chosen wisely—for example, the parameter k for k-fold cross-validation or the number of repetitions for random splitting. The different splitting algorithms used in our study are fully described in the literature. Daszykowski et al. [1] proposed a report of different rational splitting algorithms. The different effects of KS algorithm have been studied by Puzyn et al. [23]. However, to our knowledge, no comparative analysis of the three data splitting categories has been done before. No previous study has evaluated the impact of the data splitting parameter on a machine learning model for a regression problem. Therefore, in this study, we analyzed three categories of data splitting algorithms. For each algorithm, we investigated the impact of his parameter on the model generalization performance. These methods include a k-fold CV, bootstrapping [8], SPXY, and KS.

1.3 Purpose

Splitting the data into different sets is technique commonly used in machine learning. The data is usually divided into training and validation set in order to train and find the model hyperparameters (model selection) and estimate the model prediction error or accuracy. What about the algorithm used to split the data set? What is the impact of the algorithm on model accuracy or error? Some data splitting algorithms also have a parameter that needs to be optimized. For example, the parameter k for k-fold cross-validation or the number of repetitions for random splitting. This leads us to the research question of how the choice of data splitting algorithm for the training/validation set can improve the estimation of the generalization performance. The purpose of the study is to answer this research question by assuming the hypothesis that generalization performance depends on the splitting algorithm used for the model selection.

1.4 Goal

The study's primary purpose is to evaluate the quantitative differences of data splitting algorithms into both real and simulated data. Our research centered on the case where the objective of the machine learning model is to have a high correlation score. The model prediction on validation and test set are then compared based on the Pearson correlation score. The test set was unseen by the model during the training. To our knowledge, the research question proposed here has not been addressed before. [15] conducted a comparative analysis of different k-fold cross-validations.

Dobbin and Simon [5] proposed an algorithm for train set size planning. No previous studies went deeper to analyze the impact of sample splitting parameters on the machine learning model's generalization performance.

1.4.1 Benefits and Ethics

Through this study, reader can learn more about the reasoning for selecting a specific data splitting algorithm when implementing machine learning. The real data used in this study are data collected from Dream olfaction challenge website. Data were anonymized and published . No personal data revealed during our study.

1.5 Methodology

The research started by collecting previous studies nearly matching our research topic, and quantitative research methodology was used to answer the research question. Based on the literature review, we selected different sample splitting algorithms to be analyzed.

We explored different data splitting approaches. The purpose was to see which splitting will improve the model prediction on the unseen data set. A literature study was also conducted about different algorithms used to quantify the similarity between two sets of molecular descriptors.

1.6 Delimitations

There is a possibility that by evaluating more splitting algorithms, another result could be found. No implementation of new machine learning will be done to test the different splitting and verify it. Our focus is to retrain the existing model with rational splitting.

1.7 Outline

The remaining part of the report is structured as follows; in chapter two, we present the different splitting algorithms chosen for our study. Chapter three focuses on our research methodology. The result of our implementation is shown in chapter four. Finally, discussions and future work suggestions are presented in chapter five..

2 Extended Background

This section presents details explanation of the different splitting algorithms used in our work.

2.1 Data Splitting Method

2.1.1 Cross-Validation (CV)

Based on our literature review, a cross-validation algorithm is a method commonly used in machine learning. Data is split into k different parts. For each iteration, k-1 parts are used to train the model and the remaining part as a validation set. The process is iterated according to the number of fold. The generalization performance of the model is the average of the estimated scores. The model parameter with the best averaged predictive score is used as the optimal. This technique is referred to as K-fold cross-validation algorithm." More details description can be found in [12]

2.1.2 Bootstrap

The bootstrap algorithm is a method of resampling data. It is assessing the statistics and properties of a potential distribution without actually knowing its distribution [18]. The work of Kohavi [19] has proven that bootstrap is a good resampling method for selecting a machine learning model. Bootstrap randomly selects a subset of samples (with replacement) to fit and train the model and the remaining subset to validate the model. This process is repeated many times. After each process, the predictive performance of model is estimated with the validation set. The average of the scores is considered as the final estimation of the model generalization performance.

2.1.3 Kennard-Stone and SPXY sampling algorithms

Kennard-stone algorithm is probably the best-known method of uniform design among molecular modeling practitioners. The algorithm selects a representative subset according to relatively simple rules that can be summarized in the following steps [1]:

- select object closest to the mean.
- select object that is the most dissimilar to the first
- select object that is the most dissimilar to its nearest object already belongs to the subset
- stop it the subset contains the desired number of objects

The detailed implementation often differs from the general algorithm described previously. There are different measures of dissimilarity, ranging from Euclidean distance to the Tanimoto coefficient that can be used [1]. The DUPLEX algorithm is a modification or extension of the algorithm published by Snee [6]. The algorithm is used to create two subsets (training and test) that have similar statistical properties. Some further often applied subset selection approaches are sphere exclusion [8], OptiMism [4], and D-optimal design [17]. In our study, the Kennard-Stone method is implemented following the description published in [11] with Euclidean distance as a metric, Eq(1) . SPXY [7] algorithm is similar to the KS algorithm. The main difference is that SPXY took both X and Y variables into account when calculating the distance between samples, Eq(2,3). Assume we have sample matrix as:

$X\ =\begin{pmatrix} X11\ldots\ X1n \\ \ldots \\ Xn1\ldots\ Xnn \end{pmatrix}$Y = $\begin{pmatrix} Y1 \\ \ldots \\ Yn \end{pmatrix}$ The core of KS and SPXY algorithms are maximum-

$$d_{\mathbf{x}}(p,q)=\sqrt{\sum_{j=1}^{J}[x_p(j)-x_q(j)]^2};\quad p,q\in[1,N] \qquad (1)$$

$$d_y(p,q)=\sqrt{(y_p-y_q)^2}=|y_p-y_q|;\quad p,q\in[1,N] \qquad (2)$$

$$d_{\mathbf{x}y}(p,q)=\frac{d_{\mathbf{x}}(p,q)}{\max_{p,q\in[1,N]}d_{\mathbf{x}}(p,q)}+\frac{d_y(p,q)}{\max_{p,q\in[1,N]}d_y(p,q)};\quad p,q\in[1,N] \qquad (3)$$

minimum distance split, and we can define another distance metric according to the real situation. Euclidean distance metric was used in this study.

3 Research methodology

The choice of the research methodology is a important steps in a research project. The research method, approach, and strategy used in the study are based on the choice of specific research methodology.

3.1 Choice of research method

A methodology is essential when answering a research question. The choice of a research method depends on the study goal or the expected outcome [14]. A research method can be divided into a qualitative or quantitative group. A qualitative method is used to understand opinion, meaning, and usually performed with small data enough to reach good [14]. On the other hand, ”A quantitative method formulates facts and uncovers patterns in research based on measurable data analysis. The method is usually performed on large datasets with statistics tools to test the hypothesis” [14]. In addition to the research method, different approaches exist to allow us to draw a conclusion of a study. The most common research approaches are abductive, inductive, and deductive. When conclusions are drawn based on knowledge from know assumptions, it is called a deductive approach. For an inductive approach, conclusions are drawn from observation or presented facts, generally used for qualitative methods [9]. When deductive and inductive approaches are combined during the research to conclude, it is called the abductive approach [14].

We used a quantitative method with a deductive approach in our study for the following reason. First, we needed to compare the different splitting algorithms based on the numerical result of our machine learning. Second, with a deductive approach, we can draw conclusions based on the generalization score of the machine learning model built based on the splitting algorithm.

3.2 Application of research method

3.2.1 Data Collection

Simulated data

In this study, simulated data were used to compare the different splitting algorithms. We generated a regression problem base on Friedman one (1) of R package. The algorithm is described by Friedman [1] and Breiman [2]. We simulated different datasets of size,100, 500, and 1000. Friedman algorithm gives ten independent features uniformly distributed on the interval [0, 1]. We also generated data of 1000 samples. This data set is used as a test to evaluate the model performance on unseen data.

Chemoinformatic features of molecules

The real data used in this study is the molecular dataset collected from the dream olfaction challenge website. It consists of 476 structurally diverse odorant molecules. Among the molecules, we can found cyclic molecules, organosulfur molecules, and ester molecules. We generated molecular features with the Dragon software version 6 [34] . Each molecule has 4,884 different chemical features. In our study, the molecular chemical features were used to predict odor intensity.

The perceptual rating of the odor intensity was originally collected during the smell study[34]. The rating varied between 0 and 100, where 0 means ”extremely weak,” and 100 is ”extremely strong” intensity. Sixty-one people rated the differents molecules without olfaction training. Each molecule was assigned to each subject at high and low concentration. Twenty (20) molecules were tested twice. We have 992 stimuli or data points (476 plus 20 replicated molecules at two different concentrations). The dataset of 476 chemicals was already divided into two subsets 407 for the training set and 69 for the test set. We used the same test set. However, the train set was splitting again into training/validation for building the machine learning model.

3.2.2 Preprocessing of the real data

Data Preprocessing is a technique to clean and prepare data for statistical analysis. For the molecular features, we have removed the columns with constant values or NaN values. After data cleaned, we end up with 3085 features. The molecular input features were also normalized to values between 0 and 1. The formula is given as:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$ x is the original value and x' the scaled one .

3.3 Generate a random regression problem

Simulated data are widely used to assess optimization methods. This is because of their ability to evaluate certain aspects of the methods under study; these aspects are impossible to look into when using real data sets due to the limited size of data available. The simulated datasets were generated by using the first Friedman function in the R package. It generates a regression problem with samples of 10 dimensions randomly sampled.The algorithm is described by Friedman [1] and Breiman [2]. The function gives two targets Y values, One with noise N(0,1) and the true values (without noise). This allows us to compute the expected correlation score of the regression problem.

3.4 Software

Python is one of the topmost languages. It is used primarily for performing data analysis. The flexibility and the extended libraries make it more used in a data science project. All the code was written with python 3.6. All the different machine-learning algorithm was built with Sklearn library. Other python libraries, Pandas, Numpy, Scipy, Matplotlib, were used for data manipulation.

3.5 Implementation

Random forest, a popular ensemble learning algorithm for regression and classification [32], was used and applied to simulated and real data. The random forest model has hyper-parameters that need to be optimized. In this study, We have investigated the following four parameters through different splitting algorithms:

- n_estimators: number of trees in the forest. We look for the optimal parameter in the interval [10, 2000].
- max_depth: maximum depth of each tree. We have set an interval of 10 to 110 to search the optimal value. By default, maximum depth is set to None. In this case, each tree of the forest will expand until every leaf is pure.
- max_features: The max_features parameter specifies the size of the random subsets of features to consider when splitting a node.
- ”min_samples_leaf: minimum number of samples needed at a leaf node. By default, we have one (1) sample. The internal range of search was [1-4]” [13].

For each data sampling algorithm (K-fold, Ken-Stone, SPXY, and bootstrap), a parameter is used to divide the data into two subsets: training set and validation set. And we fine-tuned the model to find an optimal random forest model parameters. Below is the list of sampling algorithms and the parameters used in this study:

- Cross-validation: the number of folds was 3, 5, and 10.
- Bootstrap: the number iteration was 10, 50, and 100.
- KS and SPXY: we selected as training set the 10, 20, and also 80% of top-ranked samples.

We tested many parameters to evaluate the effect of data splitting algorithm parameter on model selection. Let consider the case of SPXY algorithm. If the parameter is set to 10%, with a dataset of 100 samples, the training set would only have ten (10) samples, and the validation set 90 samples. We find-tune the machine learning model with a set of hyperparameters to find the optimal model. Once the optimal model was found, the model is retrained with the full data set. The generalization performance is estimated on the test set. In real-world applications, test sets are usually chosen from the same dataset as the training and validation set. However, in our study, we generated an independent set. The main reason is to have a stable estimation of model performance and avoid the impact of sample size and data splitting methods. Moreover, this is important if we want to have a reliable comparison of the splitting methods and their different parameters. It can only be done when we have access to unlimited samples, such as simulated data.

4 Results

The correlation score of the different data sets is presented in 4.1. From the result, it is visible that the most important element is the size of the dataset. The Pearson correlation score (PCS) variations decrease as the number of samples increase. For data of size 1000, the Pearson correlation nearly becomes constant. With a significant representation of samples, the choice of data splitting algorithm and its setting become less critical. However, on small datasets (100 samples), there was a significant variation of the validation set PCSs. The scores were also on test sets. This highlights the importance of a proper data splitting algorithm when working with a small dataset to get the best possible model.

We noticed a significant variation within the Pearson correlation scores (PCSs) on the validation set than those for the test set, particularly with a small dataset of 100 samples. Overall, the models overfitted, the PCSs of validation sets were above those of blind sets; this is consistent with previous researches [2]. The KS algorithm showed the most significant variations in PCSs of validation sets. When a small set of samples was used as a training, the estimated correlation of the validation set was lower than the test set. However, The model overfitted when much data (>60%) were selected. The estimated correlation of the test set was lower than the validation set.

When comparing the result of the models built based on KS and SPXY on both simulated and real data, SPXY generated more over-optimistic estimations than K-S. On simulated data of size 100, SPXY reached a correlation of 85% on the validation set, when only 40% of the samples were used for training. However, K-S needed at least 60% of samples to achieve the same PCSs. When 40–60% samples were used to train the model, the gap between the validation and test sets was much smaller in terms of the PCSs. The difference between these two types of PCSs was still much more significant than k-fold cross-validation splitting. It is important to acknowledge that there are other sampling algorithms [28–30], which may perform better.

When the model was trained based on the K-fold splitting, we observed a few variations in term PCSs for validation sets. Furthermore, the variations were

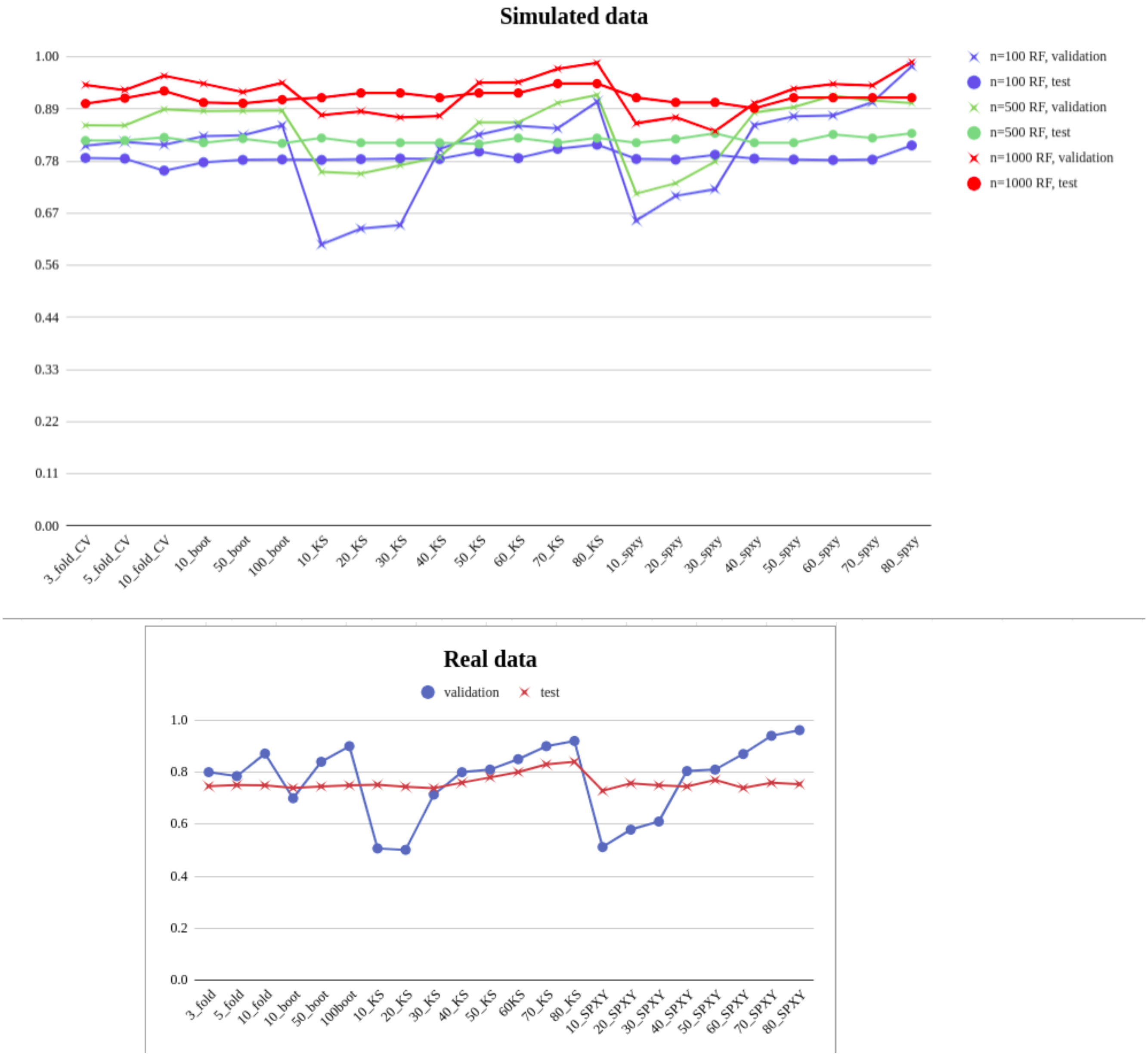

Figure 4.1: Pearson correlation scores for both simulated and real data.

still relatively low than the PCSs of the test sets. When the training data was very small or too many, the difference in terms of Pearson correlation between the validation and test set was more significant with the k-fold algorithm. This highlighted the importance of having a balanced training and validation set size for good model generalization and to avoid overfitting. When implementing a machine learning model, one may think that adding more samples in the training set will improve the model's performance. However, in real-world applications, there is no other data available excepted the external test set. This can make the external test set smaller and result in a worse scenario with a bad generalization performance.

5 Conclusion

In this study, we found that model performance improved when there was more data. The choice of data splitting is data-dependent. This resulted in the conclusion that there is no definite proof suggesting which data splitting and his parameter would give a better result than others. The choice sampling algorithm would be data-dependent. Moreover, it is challenging to know which data splitting method will give a better result in advance. However, the good ratio between training, validation, and test data size can give a better generalization performance.

6 Discussion

In this study, we carried comparative analyses of data splitting algorithms for machine learning model selection. The outcomes of our research suggested that most sampling algorithms with typical parameters can have the same result; So, they are all good for model selection. Still, the high variation of the correlation score on the validation shows that the model was sensitive to the sampling algorithms, mainly with small samples of size 100.

The Pearson correlation scores of the models build based on the splitting algorithms are presented in Figure 4.1. From the result, no sampling algorithm is better than others the optimal model parameters finding. Indeed, the estimated correlation score on the test set was approximately the same for most sampling algorithms. However, when considering the difference between the validation and test score, some splitting algorithms were better. Even with the large interval of parameter settings used for every sampling algorithm, finding a splitting method, which was significantly better than the other methods, was rare. It was challenging to choose which combinations of methods and parameters were the most suitable for selecting a machine learning model. An overall impression is that with a random splitting algorithm, and a reasonable selection of the training/validation set, we can still achieve the same result as k-fold or rational splitting algorithms.

The Friedman one algorithm helped generate a regression problem with two targets output one with Gaussian noise and one with the true target value(without noise). The correlation between the two targets gives an idea of an expected score. This allowed us to evaluate the estimated generalization score with the expected one. However, the best optimal model did not reach 98% (the expected correlation score). In general, we found that PCS was higher when there was more data.

6.1 Future work

Our Suggestion for future work is to generate another type of data with different distributions; and evaluate the different algorithms with deep learning model to see if we can draw the same conclusion.

6.2 Validity and Reliability

We have followed the best practice for machine-learning project development to ensure the validity and the reliability of our solution. Indeed, for each splitting algorithm, we implemented specific machine learning model. Evaluating with one single model would have lead to another problem,the choice of parameter such as the number of tree for the random forest. So We selected the best model hyper-parameters specific to the splitting algorithm to avoid the influence of other data partitioning algorithms such as cross validation. In order to minimize the error in our code, we have used the algorithm and function from python libraries. However, the results shown are by no means exhaustive,there are other data splitting methods [24-26] may give better result.

6.3 Reproducibility

The code and dataset for rational splitting are available for use on the github. The Java software can be found here: http://sci2s.ugr.es/sicidm

References

[1] *Applied Chemoinformatics: Achievements and Future Opportunities | Chemical Informatics | Computational Chemistry & Molecular Modeling | Chemistry | Subjects | Wiley.* Wiley.com. URL: https://www.wiley.com/en-us/Applied+Chemoinformatics%3A+Achievements+and+Future+Opportunities-p-9783527806546 (visited on 09/02/2019).

[2] Boves Harrington, Peter de. “Multiple Versus Single Set Validation of Multivariate Models to Avoid Mistakes”. In: *Critical Reviews in Analytical Chemistry* 48.1 (2018). PMID: 28777019, pp. 33–46. DOI: 10.1080/10408347.2017.1361314. eprint: https://doi.org/10.1080/10408347.2017.1361314. URL: https://doi.org/10.1080/10408347.2017.1361314.

[3] Brownlee, Jason (2017-07-13). *”What is the Difference Between Test and Validation Datasets?”. Retrieved 12 October 2017.*

[4] Clark, Robert D. “OptiSim: An Extended Dissimilarity Selection Method for Finding Diverse Representative Subsets”. In: *J. Chem. Inf. Comput. Sci.* 37.6 (Nov. 1, 1997), pp. 1181–1188. ISSN: 0095-2338. DOI: 10.1021/ci970282v. URL: https://doi.org/10.1021/ci970282v (visited on 09/02/2019).

[5] Dobbin, Kevin K. and Simon, Richard M. “Sample size planning for developing classifiers using high-dimensional DNA microarray data”. In: *Biostatistics* 8.1 (Jan. 2007), pp. 101–117. ISSN: 1465-4644. DOI: 10.1093/biostatistics/kxj036.

[6] Fisher, R. A. “The Use of Multiple Measurements in Taxonomic Problems”. In: *Annals of Eugenics* 7.2 (1936), pp. 179–188. ISSN: 2050-1439. DOI: 10.1111/j.1469-1809.1936.tb02137.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x (visited on 09/02/2019).

[7] Galvao, R K H et al. “A method for calibration and validation subset partitioning”. In: *Talanta* 67.4 (Oct. 2005), pp. 736–740. URL: https://strathprints.strath.ac.uk/36685/.

[8] Golbraikh, Alexander. “Molecular Dataset Diversity Indices and Their Applications to Comparison of Chemical Databases and QSAR Analysis”. In: *J. Chem. Inf. Comput. Sci.* 40.2 (Mar. 1, 2000), pp. 414–425. ISSN: 0095-2338. DOI: 10.1021/ci990437u. URL: https://doi.org/10.1021/ci990437u (visited on 09/02/2019).

[9] Harrington, Peter de Boves. “Multiple Versus Single Set Validation of Multivariate Models to Avoid Mistakes”. In: *Crit Rev Anal Chem* 48.1 (Jan. 2, 2018), pp. 33–46. ISSN: 1547-6510. DOI: 10.1080/10408347.2017.1361314.

[10] James, Gareth (2013). “An Introduction to Statistical Learning: with Applications in R. Springer. p. 176. ISBN 978-1461471370”. In: (), p. 176.

[11] Kennard, R. W. and Stone, L. A. “Computer Aided Design of Experiments”. In: *Technometrics* 11.1 (1969), pp. 137–148. ISSN: 00401706. URL: http://www.jstor.org/stable/1266770.

[12] Kohavi, Ron. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In: Morgan Kaufmann, 1995, pp. 1137–1143.

[13] Poojari, Devesh. *Machine Learning Basics: Descision Tree From Scratch (Part II)*. May 1, 2020. URL: https://towardsdatascience.com/machine-learning-basics-descision-tree-from-scratch-part-ii-dee664d46831 (visited on 08/19/2020).

[14] *Portal of Research Methods and Methodologies for Research Projects and Degree Projects*. URL: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A677684&dswid=-2369 (visited on 09/02/2019).

[15] pubmeddev and et, Molinaro AM al et. *Prediction error estimation: a comparison of resampling methods. - PubMed - NCBI*. URL: https://www.ncbi.nlm.nih.gov/pubmed/15905277 (visited on 11/09/2019).

[16] Ripley, Brian D. *Pattern recognition and neural networks*. Cambridge ; New York: Cambridge University Press, 1996. 403 pp. ISBN: 978-0-521-46086-6.

[17] Rodionova, Oxana and Pomerantsev, Alexey. “Subset selection strategy”. In: *Journal of Chemometrics* (). URL: https://www.academia.edu/558603/Subset_selection_strategy (visited on 09/02/2019).

[18] TM, Mitchell. *Machine learning. Boston: McGraw-Hill. 1997.* ISBN: 0070428077.

[19] Trevor Hastie Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning Data Mining, Inference, and Prediction.*

[20] Westerhuis, Johan A. et al. “Assessment of PLSDA cross validation”. In: *Metabolomics* 4.1 (Mar. 1, 2008), pp. 81–89. ISSN: 1573-3890. DOI: 10.1007/s11306-007-0099-6. URL: https://doi.org/10.1007/s11306-007-0099-6 (visited on 11/09/2019).

[21] Xu, Yun and Goodacre, Royston. “On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning”. In: *Journal of Analysis and Testing* 2 (Oct. 29, 2018). DOI: 10.1007/s41664-018-0068-2.