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

Model validation is one of the important parts of building a supervised machine learning model. For creating a model with better generalization performance, one must have a sensible data splitting strategy because this is important for model validation. In this study, we conducted a comparative study on various reported data splitting methods on both real and simulated data. Our main aim in this study 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 (KS), the sample set partitioning based on joint x-y distance(SPXY), and bootstrap algorithm. These methods were applied to split the data into training and validation sets. The generalization performances estimated from the validation sets were then compared with the ones obtained from the test set.

The results from the validation sets showed that the size of the data is an important factor for a good generalization. For all the data sample methods used on small data set, the gap between the performance estimated on the validation and test set was significant. However, the gap decreased when there was more data in training/validation, and this is because the models were then moving towards approximations of the central limit theory for the simulated datasets used. Too many or few data in the training set can also lead to bad model performance. It is necessary to have the right 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

Abstract

Verifiering av modeller är en viktig del av att bygga en övervakad modell. För att skapa en modell med bättre generaliseringsprestanda måste det finnas en förnuftig datasplittringsstrategi, eftersom detta är viktigt för modellvalidering. I denna studie genomförde vi en jämförande studie av olika rapporterade metoder för att dela data och använde molekylära egenskaper för att förutsäga mänsklig luktlukt som ett exempel. Vårt huvudsakliga syfte i denna studie är att lösa problemet med hur valet av datasegmenteringsalgoritm kan förbättra uppskattningen av generaliseringsprestanda. Datasegmenteringsalgoritmen som användes i denna studie är k gånger, varianten av Kennard-Stone-algoritmen (KS) och provuppsättningspartitionering (SPXY) baserat på gemensamt x-y-avstånd. Använd dessa metoder för att dela upp data i tränings- och valideringsuppsättningar. Den uppskattade generaliseringspoängen från verifieringsuppsättningen jämförs sedan med den generaliseringspoäng som erhållits från testuppsättningen.

Resultaten av valideringsuppsättningen visar att datastorlek är en viktig faktor för god generalisering. För alla dataprovmetoder som används på små datamängder är klyftan mellan de uppskattade generaliseringsresultaten på validerings- och testuppsättningarna mycket stor. Men när det finns mer data i träning / validering reduceras gapet eftersom modellen sedan rör sig mot ungefärlig riktning för den centrala gränsteorin för den använda simuleringsdatauppsättningen. För mycket eller för lite data i träningsuppsättningen kommer också att få prestandan att minska. Det är nödvändigt att upprätthålla en lämplig balans mellan utbildnings- / valideringsuppsättningens storlek. I vår forskning är KS och SPXY split algoritmer med dålig modellprestanda. I själva verket väljer dessa metoder de mest representativa proverna för att träna modellen, medan de dåliga representativa proverna används för modellprestanda.

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

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 purposes is to build computational models with high prediction and generalization capabilities [Mitchell, 1997]. The generalization ability depends on the model complexity. Most machine learning models have one or many model parameters that are used to control the complexity of the model. The risk of over-fitting increases as the complexity of the model also increases. Over-fitting often happens when a trained model performs very well on the samples used for training but performs poorly on new unknown data, which means the model does not generalize well. Splitting the data 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 [14], which is a set of samples used to fit the parameters of the model [24]. Successively, the fitted model is challenged with the validation set. The validation set provides an unbiased evaluation of a model fit on the training dataset while tuning the model's hyperparameters [4]. This is the model selection procedure [26].

It is important to have a good performance estimation of the trained and optimized model. However, recent researches have shown that the validation set is not always enough to measure the performance of the model. Westerhuis et al. [27] have demonstrated that the performance measured by cross-validation is an over-optimistic one. Harrington et al. [13] also proved that a single split of training and validation sets could give an incorrect estimation of model performance. These studies highlight the importance of having an additional blind test set, which is not used during the model selection and validation process to have a better estimation of the generalization performance of the model. A general flowchart of a typical model validation process is given in 1.1 [28].

However, even following this process (Fig. 1), it is still not possible to tell how properly the estimated predictive overall performance of the model from the blind test set matches the real underlying distribution of the data. This is because, in real-world applications, the latter usually is unknown. Also, the estimated performance of the model can be affected by many factors such as the modeling algorithm, the overlap between the data, the number of samples available for training, and perhaps most importantly, the method used for splitting the data. This study proposes a comparative analysis of different splitting algorithms for estimating the generalization performance of the regression model on both simulated and real datasets.

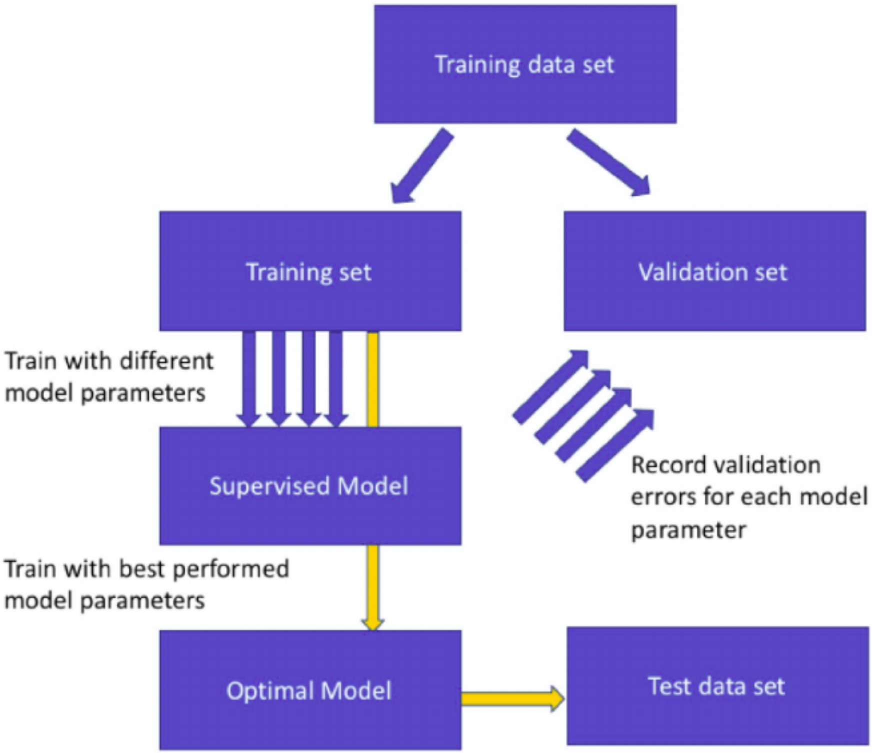
Training data set
Training set
Validation set
Train with different model parameters
Supervised Model
Record validation errors for each model parameter
Train with best performed model parameters
Optimal Model
Test data set

Figure 1.1: A model selection general flowchart. Blue arrows indicate the validation process while yellow arrows indicate the final training and test on blind test set process [28].

1.2 Problem

The different data splitting algorithms reported and used in the literature can be grouped into three categories:

- cross-validation (CV) [18];
- random selection of samples as a validation set and then using the remaining samples for training. This process is usually repeated many times, and the final estimation of the model performance is the average performance on validation sets of all the repeats. In our opinion, the bootstrap proposed by Efron et al. [8] is probably the best-known method.
- Based on the distribution of the data, systematically selecting a given number of the most representative samples from the datasets and using the remaining samples for validation is a third approach. Kennard-Stone algorithm (KS) [16] is a good example of such a method.

These data splitting have parameters that need to be optimized, such as the number of folds with CV and the number iterations random selection, etc. All these methods can be found in the literature. Daszykowski et al. [1] gave an excellent report of data splitting algorithms based on rational splitting for sample algorithm selection; Puzyn et al. [23] conducted a comparative analysis of the effect of KS. However, to the best of our knowledge, a comprehensive comparison of the methods across all three categories, particularly concerning the impact of choosing different parameter setting on each method on the regression problem, is still lacking. Therefore, in this study, we analyzed multiple data splitting algorithms commonly published in the literature, and we investigated a broad range of several parameter settings. These methods include a k-fold CV, bootstrapping [8], KS, and sample set partitioning based on joint x-y distance(SPXY).

1.3 Purpose

The split-sample approach is a widely used study design in high dimensional settings. This design divides the data into a training, validation, and test set as a means of finding the model hyperparameters (model selection) and estimating the model prediction error or accuracy. This research addresses the question of how the choice of data splitting algorithm for the training/validation set can improve the estimation of the generalization performance. Our study assumes the hypothesis that generalization performance depends on the splitting algorithm used for the model selection.

The purpose of the study is to evaluate the quantitative differences of data splitting algorithms into both real and simulated data. This study focused on the objective of obtaining a machine learning model with a high correlation score. The estimated generalization performances from the validation sets are then compared with the ones obtained from the blind test sets, which were generated from the same distribution but were unseen by the training/validation procedure used in model construction. The question addressed in this research has not to our knowledge been addressed before. Sample splitting has been addressed in other contexts, such as comparing different k-fold cross-validations [22] or developing hold out estimation theory [6] and bounds on Bayes error [15]. Mukherjee et al. [20], Fu et al. [10], and Dobbin and Simon [7] developed methods for planning the size of a training set. These studies do not address the effect of choosing different parameter setting on each method and the model the generalization performance.

1.4 Goal

The objective of this study is to successfully apply a rational splitting algorithm to olfaction odor prediction. The long term goal is to improve the machine learning model prediction of odor perception by choosing a splitting algorithm correctly. A model for predicting the odor of molecules will provide fundamental information about the transformation of odorants into olfactory perception in the brain. It should help to design and improve odorants or fragrances to positively influence our emotional state. Such methodologies are incredibly engaging as they are non-pharmacologic. Using smell as a positive- effect inducer, it appeared olfactory feelings diminished the pressure reactions in people and improved the subjects' mindset [19]. This features the likelihood to utilize smell to instigate and enhance well-being. Besides, being able to predict the odors of a chemical will dramatically accelerate the design of new molecules to be used as fragrances. Currently, perfume chemists synthesize many molecules to obtain a new ingredient, but most of them will not have the desired qualities.

1.4.1 Benefits, Ethics and Sustainability

Evaluating the quantitative differences of the splitting algorithm will help to understand the importance of the splitting algorithms. It will help to build a machine learning model with better accuracy.

The ethical aspect is always of high importance when introducing technological advances into people's lives and perhaps even more so when it regards their health. The collection of data can be done only with the agreement of the subject. This is why one should always verify when using a dataset that all the agreements have been collected. Furthermore, we must consider both current and future uses of data. Some subjects refused to make their data public for the challenge. Also, the data used in this research are all anonymized before making it public.

1.5 Methodology

The work began with a literature study to collect relevant knowledge necessary to proceed with the rest of this thesis project. We used a quantitative method to answer the research questions. The planned procedure for our master thesis is the following: based on the research of existing methods and metrics, an iterative knowledge discovery process will be started to answer the given research question. This process includes the choice of the algorithm, the parameter tuning, the implementation, and the model evaluation.

Different approaches to evaluating splits of the data are examined. The purpose was to see which division will improve the model prediction on the unseen data set. To select which splitting algorithm would be evaluated, a literature study was 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 following chapter 2 will address the theoretical background concerning this work, the introduction of some chemical similarity algorithm. After describing the taken methodology in chapter 3, the results of the experiments will be shown in chapter 4 and 5. Finally, chapter 6 will summarize the work and give an outlook for possible future research and expansion on this topic.

2 Extended Background

In this section, a brief description of the model used to simulate data is given, followed by a short review of all the data splitting methods used in this study.

2.1 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. This dataset is described in 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.

2.2 Data Splitting Method

2.2.1 Cross-Validation (CV)

Cross-validation is maybe the most commonly used data sample algorithm in model selection. It splits the data into k different elements. One part is used as a validation set, and the remaining k-1 parts are used to train the model. This method is repeated k times in order that each piece has been used as a validation set once. The saved predictive performances are then averaged; the optimal model parameter is determined as the one with the best averaged predictive performance. This technique is commonly referred to as K-fold CV or leave-one-out cross-validation (LOO-CV) when k is equal to the total number of samples.

2.2.2 Bootstrap

The bootstrap is a method of resampling data. It is assessing the stats and properties of a potential distribution without actually knowing its distribution [18]. It has been proved to be a good resampling method for selecting a machine learning model [19]. Bootstrap randomly selects n samples with replacement. These n samples are used as the training set and the remaining samples as the validation set. This process is repeated t times (e.g., t = 100) and the model performance on the validation sets are saved and averaged as the final estimation of the model generalization performance.

2.2.3 Kennard-Stone and Sample Set Partitioning Based on Joint X–Y Distances (SPXY) methods

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 [2]:

- 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 [2]. The DUPLEX algorithm is a modification or extension of the algorithm published by Snee [9]. 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 [12], OptiMism [5], and D-optimal design [25]. In our study, the Kennard-Stone method is implemented following the description published in [17] with Euclidean distance as a metric. SPXY [11] 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. Assume we have sample matrix as:

$$X = \begin{bmatrix} X11 & X12... & X1n \\ X21 & X22... & X2n \\ .... & & \\ Xn1 & Xn2... & Xnn \end{bmatrix} \quad Y = \begin{bmatrix} Y11 & Y12... & Y1n \\ Y21 & Y22... & Y2n \\ .... & & \\ Yn1 & Yn2... & Ynn \end{bmatrix}$$

The distance calculation formula for KS and SPXY are shown below:

$$d_{KS}(i,j) = \sqrt{\Sigma_{t=1}^{m}(x_{it} - x_{jt})^2}$$

$$d_{SPXY}(i,j) = \frac{\sqrt{\Sigma_{t=1}^{m}(x_{it} - x_{jt})^2}}{\max\limits_{i,j\in[1.n]} \sqrt{\Sigma_{t=1}^{m}(x_{it} - x_{jt})^2}} + \frac{\sqrt{\Sigma_{t=1}^{s}(y_{it} - y_{jt})^2}}{\max\limits_{i,j\in[1.n]} \sqrt{\Sigma_{t=1}^{s}(y_{it} - y_{jt})^2}}$$

The core of KS and SPXY algorithms are maximum-minimum distance split, and we can define another distance metric according to the real situation. Euclidean distance metric was used in this study.

3 Method and Methodologies

When conducting research projects, methods, and methodologies are essential to plan and steer the work to achieve a proper, correct, and well-founded result. The research methods, approaches, and strategies used in this study are based on the choice of specific research methodology.

3.1 Methodology

Quantitative research is used in this study. Indeed, the method uses measurable data to formulate facts and uncover patterns in research. The method is usually performed on large datasets with statistics tools to test the hypothesis [21]. The method allows us to do statistical analysis of our results from the different splitting algorithms. It also allows us to evaluate the performance of our model in order to verify our hypothesis. The applied research method was chosen to answer our research question. This method is used to answer specific research questions. It often based on previous research and uses data directly from real work and applies it to solve problems [21].

3.2 Research approach

The most common research approaches are abductive, inductive, and deductive [21]. The most appropriate approach for quantitative research, the methodology used in this study, is deductive.

3.3 Research Strategy

Our study has Ex post facto research characteristics since it is carried out after the data is already collected [21]. The analysis is carried out with statistics.

3.4 Data Collection

All the data used in this study are publicly available. So the required data collection method did not require in this study. However, the data collection protocol is described in the following subsection.

3.5 Dataset

3.5.1 Simulated data

In this study, artificial data was used to compare the different splitting algorithms. We generated a Friedman 1 regression problem. The dataset is described in Friedman [1] and Breiman [2]. We generated different data of size: 100, 500, and 1000 samples. The dataset wimulaas sted using the Friedman 1 function from the R package. It gives ten independent features uniformly distributed on the interval [0, 1].

The model training/validation was performed using the data splitting methods, as listed above, with a wide range of parameter settings (vide infra). The estimated model performances on the validation sets were then compared with the ones obtained from the corresponding blind test sets which were 1000 additional samples, also generated with the Friedman 1 function from the R package but, unknown to the training/validation procedure.

3.5.2 Real 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. Each molecule has 4,884 different chemical features calculated by a commercial cheminformatics software package known as Dragon(version 6) [34]. With these features, we can establish structure-odor relationships and further develop machine learning prediction models. However, in our study, the molecular chemical features were used to predict the odor intensity.

The perceptual rating of the odor intensity was originally collected during the smell study[34]. Sixty-one subjects rated the 476 molecules without olfaction training. Only 49 gave their permission to use their data. There were naive without any olfaction training. Each molecule was assigned to each subject at high and low concentration. Twenty molecules were tested twice. In total, we have 992 stimuli or data points (476 plus 20 replicated molecules at two different concentrations). The rating values are between 0 and 100, where 0 means ”extremely weak,” and 100 is ”extremely strong” for intensity. 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.

3.5.3 Prepossessing of the real data

Data prepossessing is a technique to clean and prepare data for statistical analysis. There were many cases where subjects indicated that they smelled nothing, so the intensity rating was automatically set to 0; therefore, we have removed all the NaN’ entries. 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 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 .

3.6 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.

4 Implementation

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

- n_estimators: The n_estimators parameter specifies the number of trees in the forest of the model. The value for this parameter was varied from 10 to 2000.
- max_depth: The max_depth parameter specifies the maximum depth of each tree. We have set an interval of 10 to 110 to search the optimal value. The default value for max_depth is None, which means that each tree will expand until every leaf is pure. A pure leaf is one where all of the data on the leaf comes from the same class.
- max_features: The max_features parameter specifies the size of the random subsets of features to consider when splitting a node.
- min_samples_leaf: The min_samples_leaf parameter specifies the minimum number of samples required to be at a leaf node. The default value for this parameter is 1, which means that every leaf must have at least 1 sample that it classifies. The internal range of search was [1-4].

K-fold, Ken-Stone, SPXY, and bootstrap sampling with a wide range of parameter settings were applied to split each dataset into training and validation set and used to train the models and find optimal model parameters. The parameter settings of these methods that we used are listed below:

- k-fold CV: k was set to be 3, 5, and 10.
- Bootstrap: t was set to be 10, 50, and 100.
- KS and SPXY: 10, 20, 80% of top-ranked samples in the dataset were selected as the training set.

For each splitting algorithm, we tested a wide range of parameters, to show the effect of using some unreasonable parameter settings on model selection. For data of 100 samples, KS, with the 10% top-ranked samples to be used, the training set would only contain ten (10) samples and the rest for the validation set. Once the optimal model parameters were found, the model has trained again on the full data, with training and validation set combined, using the optimal model parameter and applied to the test set to estimate its generalization performance.

Three simulated datasets are generated with 100, 500, and 1000 samples for model training and validation. Finally, a new dataset with 1000 samples was generated as a blind test set. It is important to mention that in real applications, training, validation, and test set are usually selected from the same dataset. In our study, we decided to generate an external test set with a large number of samples. The main reason is to have a stable estimation of model performance. Also, the test set will not be affected by some factors such as sample size, data splitting methods. It is important for a fair comparison over various combinations of datasets, data splitting methods, and their different parameter. This can only be done when we have access to unlimited samples, such as simulated data.

5 Results

The correlation score of the different data sets is presented in 5.1. From the result, it is visible that the dominant factor is the size of the dataset, 100,500,1000. The difference in the Pearson correlation score of both validation and test sets decreased significantly as the number of samples increased. For data of size 1000, the score obtained by using the different data splitting methods had nearly become a constant. This implies that the choice of data splitting algorithm and its parameter become less important.with a significant representation of samples, the choice of data splitting algorithm and its setting become less critical. However, on small datasets (100 samples), it happens evidently that the scores of validation sets varied very significantly, and the low PCSs on test sets were evident. This highlights the need to have a proper data splitting algorithm when working with a small dataset to get the best possible model on.

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 PCSs of validation sets were above those of blind sets, indicating that the model was overfitting; this is consistent with previous researches [3]. The KS algorithm showed the most significant variations in PCSs of validation sets. The estimated correlation was lower than those in test sets when small samples(<30% were used for training. However, The model overfitted when much data (>60%) were selected.

When comparing the result of KS and SPXY on both simulated and real data, SPXY was generating more over-optimistic estimations than K-S. On simulated data, with 100 samples, when only 40% of the samples were used for training, SPXY had achieved 85% PCSs on the validation set, while for K-S, at least 60% samples were needed to achieve the same PCSs. When 40–60% samples selected for training, the gap between the PCSs on the validation and test sets was much smaller. The difference between these two types of PCSs was still much more significant than k-fold cross-validation data splitting. It is important to mention that there are other data splitting methods [28–30], which may perform better.

For the k-fold splitting method, although these had less variation in PCSs on

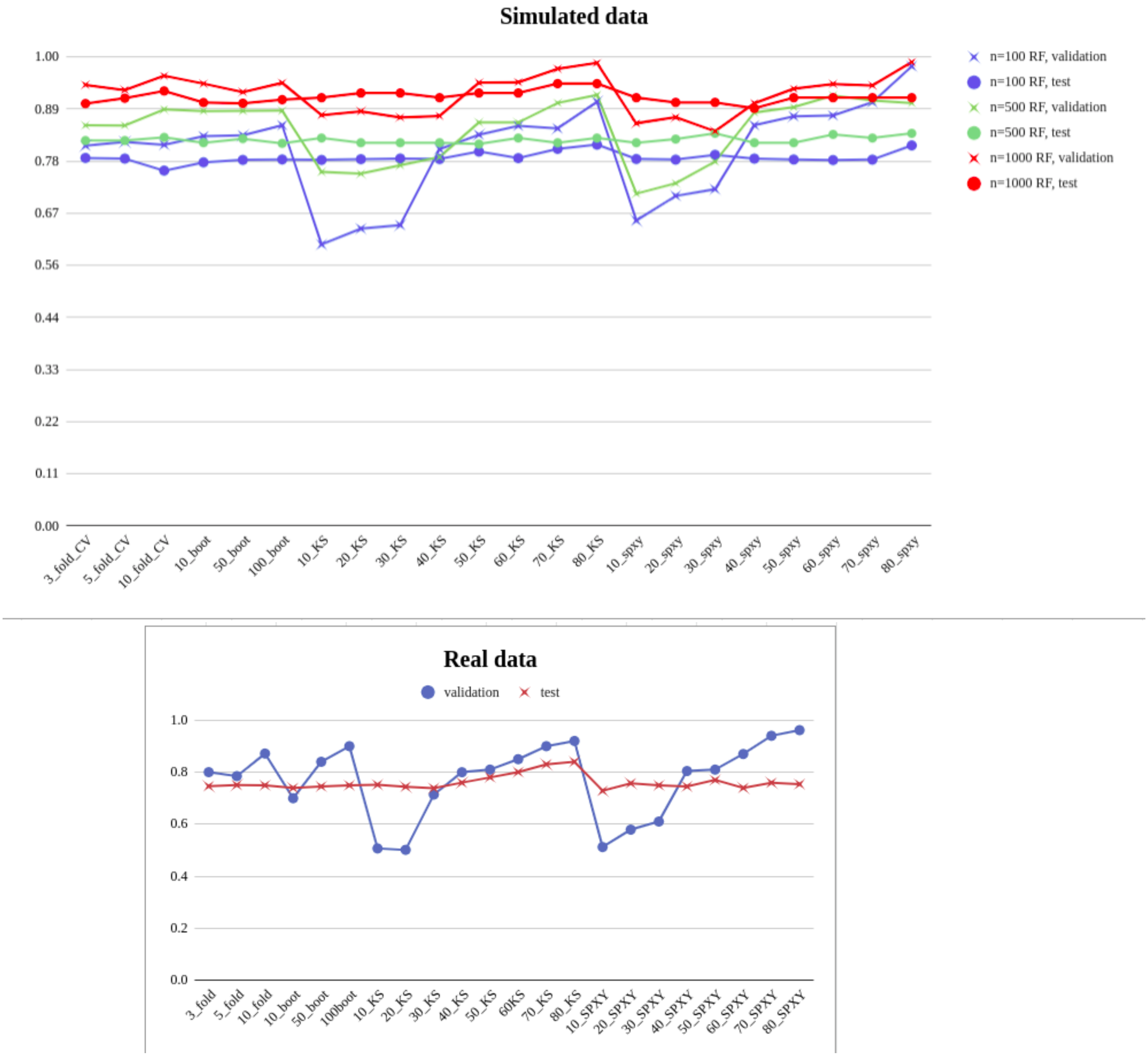

Figure 5.1: Pearson colleration scores for both simulated and real data .

validation sets, such variations were still significantly low than those of the test sets. The differences were also most visible when the data was minimal. When we had too many or very small inputs in the training set, gaps between the two validation and external test set estimated score was significant. This highlighted the importance of having a balanced training and validation set to have a good generalization and avoid overfitting. Imagine that when someone tries to build a regression model on data with significant overlaps, it is intuitive to think that the most convenient way to improve the performance of the model is to get more samples for the training set. However, in real-world scenarios where data may be hard to obtain, no other samples are available. This would consequently reduce the size of the test set and thus lead to an even worse estimation of the model's generalization.

5.1 are illustrative summaries of the Pearson correlation scores. The result shows that no data sampling algorithm had any obvious advantage over others in finding the optimal model parameters. Instead, the estimated correlation score on the test most data splitting methods with many different parameter settings was approximately the same. However, when considering the difference between the validation and test score, some splitting algorithms were better. This suggests that even with a wide range of parameter settings used for every data splitting method, finding a sample 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 model selection. An overall impression is that applying a random sampling method, repeated many times (t≥50), and a reasonable balance between training and test set (50–70% for training), we can get the same result as k-fold or rational splitting algorithm.

6 Discussion

In this study, we carried a comprehensive comparison study on different data splitting methods for model selection and validation. We used the random forest as a machine learning algorithm to build the model. Real data and simulated data were used in the research. Friedman 1 algorithm was used to generate the simulated data of 100, 500, 1000 samples. The results suggested that most splitting methods with typical parameter settings resulted in a similar correlation score 5.1; therefore, they are all viable options for model selection. However, the high variation of the correlation score on the validation was very sensitive to the data splitting method, as well as its parameter setting, mainly when small dataset with just 100 samples were used.

There is no definite proof suggesting which method and parameter combination would always provide significantly better results than others. The choices of which method to use for data splitting and which parameters to use cannot be decided a priori and would be data-dependent. However, a good balance between size training, validation, and test set can give a stable estimation of model performance.

The Friedman one model was useful as this allowed us to generate a dataset 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 allows us to compare the generalization performance estimated from the splitting algorithms with the expected one. We found that even the performance of the best model cannot reach the expected correlation score of 98%. In general, we found that model performance improved when more samples were used.

6.1 Validity and Reliability

We have followed the best practice for machine-learning project development to ensure the validity of our solution. 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.

6.2 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] /s, Doi et al. *Representative subset selection*. 2002.

[2] *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).

[3] 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`.

[4] Brownlee, Jason (2017-07-13). """What is the Difference Between Test and Validation Datasets?". Retrieved 12 October 2017." In: ().

[5] 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).

[6] Devroye L Gyorfi L, Lugosi G. A. *Probabilistic Theory of Pattern Recognition*. URL: `https://scholar.google.com/scholar_lookup?title=A+Probabilistic+Theory+of+Pattern+Recognition&author=L+Devroye&author=L+Gyorfi&author=G+Lugosi&publication_year=1996&` (visited on 1996).

[7] 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`.

[8] Efron B, Tibshirani R. *An Introduction to the Bootstrap. Boca Raton: Chapman and Hall/CRC; 1993.*

[9] 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).

[10] Fu, Wenjiang J. et al. “How many samples are needed to build a classifier: a general sequential approach”. In: *Bioinformatics* 21.1 (Jan. 1, 2005), pp. 63–70. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bth461.

[11] 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/.

[12] 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).

[13] 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.

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

[15] K., Fukunaga. *Introduction to Statistical Pattern Recognition.* (Visited on 1990).

[16] Kennard, R. W. and Stone, L. A. “Computer Aided Design of Experiments”. In: *Technometrics* 11.1 (Feb. 1, 1969), pp. 137–148. ISSN: 0040-1706. DOI: 10.1080/00401706.1969.10490666. URL: https://tandfonline.com/doi/abs/10.1080/00401706.1969.10490666 (visited on 12/31/2019).

[17] 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.

[18] Kohavi, Ron. "foAr AStcucduyraocfyCErsotsims-VatailoidnaatinodnManoddeBloSoetlsetcrtaiopn". In: (), p. 7.

[19] Lehrner, J. et al. "Ambient odor of orange in a dental office reduces anxiety and improves mood in female patients". In: *Physiol. Behav.* 71.1 (Oct. 1, 2000), pp. 83–86. ISSN: 0031-9384. DOI: `10.1016/s0031-9384(00)00308-5`.

[20] Mukherjee, Sayan et al. "Estimating dataset size requirements for classifying DNA microarray data". In: *J. Comput. Biol.* 10.2 (2003), pp. 119–142. ISSN: 1066-5277. DOI: `10.1089/106652703321825928`.

[21] *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).

[22] 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).

[23] Puzyn, Tomasz et al. "Investigating the influence of data splitting on the predictive ability of QSAR/QSPR models". In: *Struct Chem* 22.4 (Aug. 1, 2011), pp. 795–804. ISSN: 1572-9001. DOI: `10.1007/s11224-011-9757-4`. URL: `https://doi.org/10.1007/s11224-011-9757-4` (visited on 12/31/2019).

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

[25] 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).

[26] Tibshirani, Sami and Friedman, Harry. "Valerie and Patrick Hastie". In: (), p. 764.

[27] 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).

[28] 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.