MONTAN
UNIVERSITÄT
LEOBEN

Chair of Subsurface Engineering

Master's Thesis

Use and limitations of various metrics to assess the quality of extreme sparse datasets in geotechnics

Matthias Hahn, BSc

March 2023

FUTURISM
"Of course, there's no way to prove we're living in a simulation."
LK+LM

Source:[15]

Abstract

In data science and statistics, metrics are the measures of a quantitative assessment of dataset(s). In machine learning (ML), metrics are used to monitor the performance of a model during training and testing (therefore sometimes called “performance metrics”) by calculating a distance between predicted and true outputs. All ML models need a metric to access the model’s accuracy in mapping the inputs X to the outputs y.

The ML task can be classification or regression, so the performance metrics. Classification is a supervised learning method that predicts qualitative responses. A classification problem requires that examples are classified into a finite number of classes. Thus, classification is mapping the input variables to discrete output variables. Regression is a supervised learning method used to determine the relationship between independent variables X and dependent variable(s) y. The regression model is mapping input variables to a continuous output variable(s). There are several metrics for both problems. To mention a few:

- regression metrics: mean absolute error, mean squared error, root mean squared error, R2;
- classification metrics: accuracy, precision/recall combinations, AUROC (area under receiver operating characteristics curve).

To understand how close the results are to the objectives in Research and Development projects, choosing an appropriate evaluation metric for each class of ML is crucial. In geoengineering, the datasets often exhibit extreme sparsity and observations frequency (e.g., rare events). Therefore, the application of both ML tasks on such data requires special preprocessing (e.g., under-sampling, over-sampling, compressing). After ML models are trained on preprocessed data, their output shall be evaluated using a metric that provides the most comprehensive evaluation of the results.

MONTANUNIVERSITÄT LEOBEN
www.unileoben.ac.at

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu "Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 13.02.2023

M. Hahn

Unterschrift Verfasser/in
Matthias Hahn

Contents

1 Introduction

1.1 Machine Learning Algorithms

Machine Learning is applied when it is difficult or impossible to solve a problem with conventional algorithms. This is often the case for applied problems in nature like in biology or geology, where huge amounts of often strongly varying data are given to calculate a model. In geology these data vary vertically and horizontally in the subsurface and are often spatially limited (in geotechnics e.g. core plugs) or as overall time series (in geophysics e.g. seismic logs). With the implementation of data banks in geoscience and collecting huge amounts of information over decades, machine learning (ML) becomes a possibility for data mining to get new or more knowledge out of these data. ML can be a future possibility to obtain the first real-time models from logging while drilling or seismics during operation of tunnel boring machines. [21]

In ML the goal is to create a model out of the measured input and output of a natural given system, often without knowing exactly the physical models in the system are. Figure 1.1 shows how the loop of a supervised ML task works. S describes the natural system, x is the input and y the answer of the system. Here $f_{(x)}$ is the function which describes the real system. The learning machine LM tries to create a model which is able to predict an output y_e that is the same or sufficiently similar to y. To perform this task the LM must learn from the real data and so needs the input and the corresponding output. After the learning, the LM should be possible to predict the output like the real system or in a sufficiently accurate way. Here three main problems came up with ML: first no one can guarantee that the LM always produces the exact or sufficiently accurate answer like the real system S. Second it is difficult to use analytical methods for comparing $f_{e(x)}$ and $F_{(x)}$. Third the LM can perform well for a lot of examples and then fail at some points. [25]

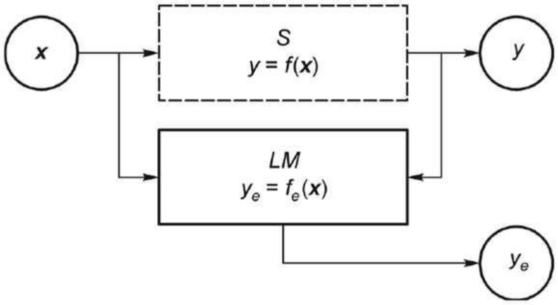


Figure 1.1: The general learning loop of a supervised learning machine LM. S is the natural system, x is the input, y is the ouput of S, y_e the predicted output of LM (Source: [25])

Like all models, LM have errors and can not predict the output exact like the real system. To get an idea how well an applied ML algorithm works, performance measuring is needed. Different ML algorithm categories are known and for each different performance measuring methods, also called metrics, have been developed in the last decades. ML is categorised in four types:

- Supervised learning
- Semi-supervised learning

- Unsupervised learning
- Reinforced learning

In contrast to unsupervised learning, supervised learning LM is fed with the input and the output of the natural system and the operator modifies the supervised LM until the predicted/calculated output is accurate enough. This thesis will only utilise supervised learning. Three subcategories are known in supervised learning:

- Classification
- Regression
- Forecasting

In classification every data point in a given data set becomes a label and is categorized. Classification algorithms differ by the amount of classes which exists and the amount of labels one data point can be given. The best example for classification ML is the email filtering algorithm for "spam" mails. Here the LM classifies the emails by checking different aspects of the mail content. In regression the LM tries to produce a continuous output value, which can be used numerically. Here, for example, the output can be the price of a house, say 220.000 Euros. In classification this house would be labeled a specific value class like 200.000-250.000 Euros or which income class is interested in this house[19]. This thesis will utilise classification and regression.

1.2 Classification Metrics

Classification is used when the data can be categorized into classes. This task can be used in geoscience for labeling rocks into lithologies based on the content of differente minerals. In classification different types of algorithms are known. The mosts common is binary classification. Here the data become only one of two possible labels. Multi-class classifiers apply one label of more than two possible labels each data point. Multi-label classifiers are able to give more than one label to every data point. Before the metrics are explained, some basic values and wordings must be described. For the class on which the performance measuring is done, the labeled data to this class are called positive, whereas the data labeled to the other classes are called negative. The feature space is the defined room where all the inputs are located. Output y of a classifier is the label of the input X. The training tries to minimize the error rate on the training data and testing simulates a practical application scenario. To obtain a perfect classifier, an infinite number of testing samples must be given, so the LM gives the same output as the natural system. In reality only a finite number of samples is given, and here the operator must decide which samples are used for training and which for testing. [25] describes three evaluation methods:

- Re-substituting test
- Independent dataset test
- Cross validation test (here the Jackknife test and n-fold test)

For the independent dataset tests the given dataset is randomly partitioned into two parts - 70% of the data for training, the rest for testing. The problem is that for small sample numbers, one can get a large variance. To minimize this problem one can run the test several times and use the average performance as an indicator. This can be unsatisfactory. The re-substituting test takes the same data for training and testing. This leads to an overestimated performance of the test and is least recommended. For limited data sets, the cross validation test helps to overcome these problems. This method can be divided into the the leave-one-out cross validation and the n-fold cross validation. At n-fold, also all data are used for training and testing. The difference is, that the dataset is divided into n parts of equal size. Each part is picked once for testing, and for the other rounds it is part of the training data. So in the k round, part k for testing, and n-1 parts for training. The metric is figured out by averaging over the whole data set. Normaly n varies betwenn 3-5. Figure 1.2 shows the n-fold testing for n=5. For this kind of testing, the predicted and thr real results have a negative correlation. This can also happen, even when there is no relationship between them. This increase with increasing n. The Jackknife test, or also called the Leave-One-Out cross validation test, uses for n the number of data points given. Here the negative correlation problem becomes overwhelming. [25]

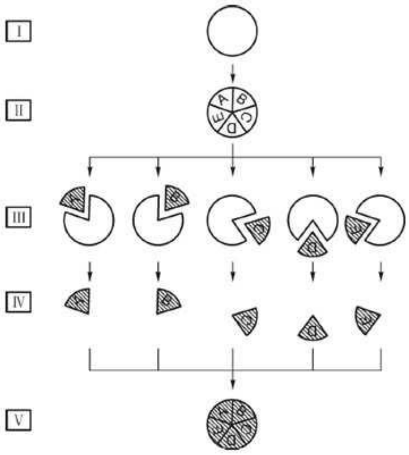


Figure 1.2: Dataset split into 5 parts equal size, A-E. Every part is used one time for testing and four times for training.(Source: https://scikit-learn.org)

Real Positive (RP) and Real Negative (RN) describes the label which the natural system would classify. Predicted Positive (PP) and Predicted Negative (PN) is the output of the LM. True Positive (TP) and True Negative (TN) are the data which the predicter labeled the same as the natural system. False Positive (FP) and False Negative (FN) are the incorrectly labeled data by the LM. The context between these values is often shown in a confusion matrix, like in Table 1.1.

	Predicted Positive	Predicted Negative
Real Positive	True Positive	False Negative
Real Negative	False Positive	True Negative

Table 1.1: An example of a confusion matrix

The most common binary metrics are the Sensitivity Equation 1.1 (also True Positive Rate; Recall), the Specificity Equation 1.2 (also True Negative Rate; Inverse Recall) and the Accuracy Equation 1.3:

$$Sen = \frac{TP}{TP+FN} \tag{1.1}$$

$$Spec = \frac{TN}{TN+FP} \tag{1.2}$$

$$Acc = \frac{TN+TP}{TN+FN+TP+FP} \tag{1.3}$$

These three equations are the most common performance measures in classification. Sensitivity gives the rate of the correct positive samples to all real positive samples. So the grade how well the predictor identifies positive samples. Specificity is the same for negative samples. The accuracy measures the ability of the predictor to correctly lablel all samples, independent of the class. These show a general information about the performance of the LM. Often users are only interested in the information of positive hits. Here the Positive Predictive Value Equation 1.4 (PPV, also Precision), the False Discovery Rate Equation 1.5 (FDR) and the Jaccard Index Equation 1.6 are useful for this case.

$$PPV = \frac{TP}{TP+FP} = \frac{TP}{PP} \tag{1.4}$$

$$FDR = 1 - PPV = \frac{FP}{TP+FP} = \frac{FP}{PP} \tag{1.5}$$

$$J = \frac{TP}{TP+FP+FN} = \frac{TP}{n-TN} \tag{1.6}$$

If the TP should be weighted stronger, it can be multiplied by 2 in the denominator and in the numerator. This is called the F_1-score.[25]

Sometimes datasets are imbalanced. Imbalanced datasets have a significant higher number of datapoints in one class than in the others. For example, in an automated picture classification of the rework of waste, it is useful to get out lithium battery cells because of the danger of fire. 99% of the pictures contain no battery, so they are true negatives. Only 1% are true positives and so the training data is imbalanced. To reduce this problems in the value of performance measuring, the Balanced Accuracy Equation 1.7 (BAcc) and the Mathhew´s Correlation Coefficient Equation 1.8 (MCC) are implemented, where -1$\leq$ MCC$\leq$ +1.

$$BAcc = \frac{1}{2} * (Sen + Spe) = \frac{1}{2} * (\frac{TP}{TP+FN} + \frac{TN}{TN+FP}) \tag{1.7}$$

$$MCC = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} = \frac{TP+TN-FP*FN}{\sqrt{PP*PN*RP*RN}} \tag{1.8}$$

are implemented, where -1$\leq$ MCC$\leq$ +1. The MCC is used especialy when the Jackknife test criterion is applied on imbalanced data. The BAcc is useful for solving most of the bias problems in imbalanced data. The MCC is equal to the Chi-Square test on the confusion matrix. Sometimes predictors are coded that they give a score to the samples and then label the samples according to a pre-defined cut off value. The decision of the cut off value strongly influences the given metrics. To eliminate this problem, the Receiver Operating Characteristic (ROC) curve method is introduced. The relation between the False Positive Rate Equation 1.9:

$$FPR = 1 - Spe = \frac{FP}{FP+TN} \tag{1.9}$$

and the Sensitivity is described using the ROC curve. Figure 1.3 shows a possible ROC curve. Good performance is shown, when the curve converges to the upper left corner. When the curve is close to the dashed diagonal, the predictor works randomly. The Area Under the Curve (AUC) indicates the probability that a randomly picked positive sample gets a higher score than a randomly picked negative sample. The form of the ROC curves can be strongly influenced by highly imbalanced datasets. [25]

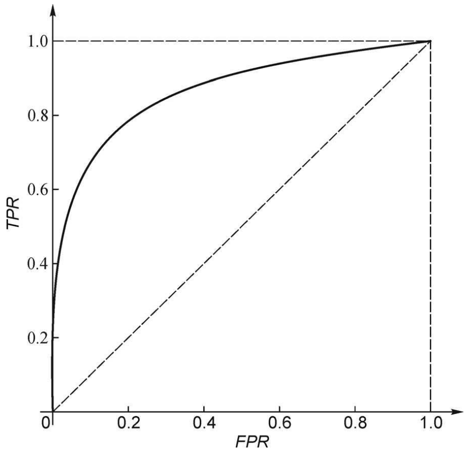


Figure 1.3: The Receiver Operating Characteristic (ROC) curve is independly of pre-defined cut off values.(Source: [25])

For the problem of the impact of the ROC curve by imbalanced datasets, the Precision-Recall (PR) curve is useful. The difference with the ROC is that sensitivity/recall and precision is used for the coordinates. So the ROC and PR curve are related to each other since every dot on a ROC curve represents a confusion matrix which leads to the sensitivity and precision for the PR curve. So the AUC of ROC and PR curves always show the same results. [25]
Until now metrics for binary-class classifier have been discussed. A commonly metric for multi-class classifiers is the overall accuracy Equation 1.10:

$$Acc_{Overall} = \frac{1}{n}\sum_{j=1}^{m} TP_j \tag{1.10}$$

where n is the total number of testing samples, m the total number of classes and TP$_j$ the number of true positives of the j-th class. Here all classes of the predictor are accounted for by one value. [25]
For multilabel-predictors the performance measure is given by aiming Equation 1.11 (multi-label PPV), by coverage Equation 1.12 (multi-label coverage), by accuracy Equation 1.13 (multi-label accuracy), by absolute-true-rate Equation 1.14 and by absolute-false-rate Equation 1.15:

$$Aim = \frac{1}{n}\sum_{k=1}^{n} \frac{|\, y_e \cap y \,|}{|\, y_e \,|} \tag{1.11}$$

$$Cov = \frac{1}{n}\sum_{k=1}^{n} \frac{|\, y_e \cap y \,|}{|\, y \,|} \tag{1.12}$$

$$Acc = \frac{1}{n}\sum_{k=1}^{n} \frac{|\, y_e \cap y \,|}{|\, y_e \cup y \,|} \tag{1.13}$$

$$ATR = \frac{1}{n}\sum_{k=1}^{n} \delta_k \tag{1.14}$$

$$AFR = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{m}[| y_e \cap y | - | y_e \cup y |] \tag{1.15}$$

Here δ_k is an indicator function, which is 1 when the predicted output is completely correct, otherwise it is 0. This means that for a multi-label predictor, all given labels must be completely correct. If the predictor gives labels to the sample, and it is incorrect, it is called over-predicting. If the predictor does not label the sample although it is true, it is called under-predicting. The ATR describes the frequency of correct predictions. The AFR is the average rate of incorrectly predicted labels. These values are calculated over-all classes and describe the performance of the predictor over-all classes. [25]

[10] provides some additional metrics. The authors describe it as quality performance metrics for one input-class and two output classes. Here the input class is the class for which the predictor is trained during the training sequence. The output class describes the unknown samples which are used for the testing. One can decelerate this as binary classifier. The wording was choosen in a more general way.

The Yoouden´s index Equation 1.16 describes the proportion between correct and wrongly classified samples:

$$YOUD = Sen - (1 - Spec) \tag{1.16}$$

The positive Likelihood ratio Equation 1.17 and negative Likelihood ratio Equation 1.18 describe the ratio between the agreement in class A and the errors in class not A and visa a via:

$$LR(+) = \frac{Sen}{1 - Spec} \tag{1.17}$$

$$LR(-) = \frac{1 - Sen}{Spec} \tag{1.18}$$

Classification Odds Ratio Equation 1.19 combine metrics like FPR and FNR, Sen and Spec in one parameter:

$$COR = \frac{LR(+)}{LR(-)} \tag{1.19}$$

Discriminant power Equation 1.20 is called "test effectiveness" and shows the possibility of the classifier to distinguish between of objects of both classes (A and not A):

$$DP = \frac{\sqrt{3}}{\pi} * (\log\frac{Sen}{1 - Sen} + \log\frac{Spec}{1 - Spec}) = \frac{\sqrt{3}}{\pi}\log COR \tag{1.20}$$

where DP>1 is OK and >1.5 is good. The Area under the receiver operating curve Equation 1.21 is already described before and can be calculated in this way:

$$AUC = \frac{Sens + Spec}{2} \tag{1.21}$$

The Gini coefficient Equation 1.22 response the errors during classification and at low values a good performance of the classifier in this field:

$$Gini = (2 * AUC) - 1 \tag{1.22}$$

The G-mean Equation 1.23 can be called the "geometrically averaged accuracy" and shows an equilibrium between agreements and errors of the predictor:

$$GM = \sqrt{Sen * Spec} \tag{1.23}$$

[10] claimed, that for a predictor with two or more input classes the above given wording of positive and negative does not work any more. Instead the terms "agreement" and "error" values are used. A value applied and belonging to class A is a_A, applied to A but belonging to B is $e_{B,A}$, applied to B but belonging to A $e_{A,B}$ and applied to B and belonging to B is a_B. T describes the total number of validation standards, TC_A and TC_B the total number of standards of classes A and B Summing up brings:

$$\begin{aligned} TC_A &= a_a + e_{A,B} \\ TC_B &= e_{B,A} + a_B \\ AC_A &= a_a + e_{B,A} \\ AC_B &= e_{A,B} + a_B \\ T = AC_A + AC_B &= TC_A + TC_B \end{aligned}$$

The Chance Agreement Rate Equation 1.24, also known as the chance-no-error rate, shows the ratio of agreements due to chance:

$$CAR = \frac{(TC_A * AC_A) + (TC_nA * ACnA)}{T^2} \tag{1.24}$$

The Chance Error Rate Equation 1.25, also the random error rate, shows the the possibility of an error when labeling is performed by chance:

$$CER = \frac{2 * TC_A * TC_nA}{T^2} \tag{1.25}$$

The Kappa coefficient Equation 1.26 describes the ratio of real agreements by considering that some of them could be assigned by chance:

$$KAPPA = \frac{Acc - CAR}{1 - CAR} \tag{1.26}$$

The Bayes´ conditional probability uses the nomenclature P(assigned class | reference class). Here all the values are used to show the probability how one label is right or wrong given.

$$P(A|A) = \frac{Prev * Sen}{Prev * Sen + (1 - Prev) * (1 - Spec)} \tag{1.27}$$

$$P(nA|nA) = \frac{(1 - Prev) * Sen}{(1 - Prev) * Sen + Prev * (1 - Sen)} \tag{1.28}$$

$$P(nA|A) = \frac{(1 - Prev) * (1 - Spec)}{Prev * Sen + (1 - Prev) * (1 - Spec)} \tag{1.29}$$

$$P(A|nA) = \frac{Prev * (1 - Sen)}{(1 - Prev) * Spec + Prev * (1 - Sen)} \tag{1.30}$$

where the prevalence is:

$$Prev = \frac{TC_A}{T} \tag{1.31}$$

which can be also expressed as the ratio of all values which actually belong to class A to all given samples. The problem with the probability performance measurement is the prevalence. [10] also describes metrics for more than two output classes. The values are like for the two output predictor, but over-all classes. The overall Agreement Rate Equation 1.32, the overall Error Rate Equation 1.33, the overall chance agreement rate Equation 1.34, the overall Chance Error Rate Equation 1.35, and the overall Kappa coefficient Equation 1.36:

$$overallAR = \frac{\sum a_i}{T} \tag{1.32}$$

$$overallER = \frac{T - \sum a_i}{T} = \frac{\sum e_i}{T} = 1 - overallAR \tag{1.33}$$

$$overallCAR = \frac{\sum TC_i * AC_i}{T^2} \tag{1.34}$$

$$overallCER = \frac{\sum TC_i * (T - TC_i)}{T^2} \tag{1.35}$$

$$overallKAPPA = \frac{overallAR - overallCAR}{1 - overallCAR} \tag{1.36}$$

[22] describes hierachical classifiers as predictors, which label the input with one class, which is either divided into subclasses or grouped into superclasses. Thus is often used for text and bioinformatic predictors. Here metrics like Precision, Recall and Fscore can be used like for binary classifiers. Invariance is used in classification to describe the independence of a metric from changes in the confusion matrix. For example, the independence of Precision or Recall of True Negative is beneficial in text classification but adverse in human communication. [22] describes eight invariances measures for binary confusion matrices, because this can be applied to other classifiers which are derived from the matrix.
I_1: metrics, which stay constant under the exchange of positives and negatives.

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix} \tag{1.37}$$

The metrics of this invariance are not trustworthy for comparing of classifiers with different or unbalanced class distributions.

I_2: metrics, which stay constant under change of True Negative counts.

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \tag{1.38}$$

I_3: metrics, which stay constant under change of True Positive counts.

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \tag{1.39}$$

I_4: metrics, which stay constant under change of False Negative counts.

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \tag{1.40}$$

I_5: metrics, which stay constant under change of False Positive counts.

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \tag{1.41}$$

I_6: metrics, which stay constant under constant change of positives and negatives.

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} k_1 * TP & k_1 * FN \\ k_1 * FP & k_1 * TN \end{bmatrix} \tag{1.42}$$

I_7: metrics, which stay constant under constant change of False Negative and True Negative, where $k_1 \neq k_2$

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} k_1 * TP & k_2 * FN \\ k_1 * FP & k_2 * TN \end{bmatrix} \tag{1.43}$$

I_8: metrics, which stay constant under constant change of False Positive and True Negative, where $k_1 \neq k_2$

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} k_1 * TP & k_1 * FN \\ k_2 * FP & k_2 * TN \end{bmatrix} \tag{1.44}$$

For the above described invariance measures, Table 1.2 shows the depending metrics. It describes with + an invariant and with - a non-invariant metric.

	I1	***I2***	***I3***	***I4***	***I5***	***I6***	***I7***	***I8***
Binarcy Classification								
Accuracy	+	-	-	-	-	+	-	-
Precision	-	+	-	+	-	+	+	-
Recall/Sensitivity	-	+	-	-	+	+	-	+
Fscore	-	+	-	-	-	+	-	-
Specifity	-	-	+	+	-	+	-	+
AUC	-	-	-	-	-	+	-	+
Multi-class classification								
Average Accuracy	+	-	-	-	-	+	-	-
Error Rate	+	-	-	-	-	+	-	-
Precision text classifier	-	+	-	+	-	+	+	-
Recall text classifier	-	+	-	-	+	+	-	+
Fscore text classifier	-	+	-	-	-	+	-	-
Precision multi class classifier	-	+	-	+	-	+	+	-
Recall multi class classifier	-	+	-	-	+	+	-	+
Fscore multi class classifier	-	+	-	-	-	+	-	-
Multi-topic classification								
Exact Match Ratio	-	-	-	+	+	-	-	-
Labelling Fscore	-	+	-	-	-	+	-	-
Retrieval Fscore	-	-	-	-	-	+	-	-
Hamming Loss	+	+	+	-	-	+	-	-
Hierarchical classification								
Precision subclass	-	+	-	+	-	+	+	-
Recall subclass	-	+	-	-	+	+	-	+
Fscore subclass	-	+	-	-	-	+	-	-
Precision superclass	-	+	-	+	-	+	+	-
Recall superclass	-	+	-	-	+	+	-	+
Fscore superclass	-	+	-	-	-	+	-	-

Table 1.2: Invariance condition for different classification metrics. + describe invariance, - non-invariance (Source: [22])

[**?**] describes the possibility of comparing two classifiers, which are trained on the same data set. The McNemar´s test divides the data set S into a training set R and a test set T. The statistical null hypothesis H_0 says, that both algorithms should have the same error rate. The test is based on a χ^2 for goodness-of-fit. The expected counts under the H_0 hypothesis are described in Table 1.3:

n_{00}	$\frac{n_{01}+n_{10}}{2}$
$\frac{n_{01}+n_{10}}{2}$	n_{11}

Table 1.3: Expected counts of the H_0 hypothesis. (Source: [**?**])

with the contingency in Table 1.4

Number of examples misclassified by both classifiers n_{00}	Number of examples misclassified by classifier A (f_A) not by classifier B (f_B) n_{01}
Number of examples misclassified by f_B but not by f_A n_{10}	Number of examples misclassified by neither f_A nor f_B n_{11}

Table 1.4: McNemar´s test contingency table (Source: [**?**])

The s value is used to describe the statistic of the χ^2 with 1 degree of freedom:

$$s = \frac{(n_{10} - n_{01} - 1)^2}{n_{10} + n_{01}} \tag{1.45}$$

If $|s| > \chi^2_{1,0.95}$, describing a probability smaller than 0.05, the H_0 hypothesis can be rejected and the two algorithms have different performance. The McNemar´s test has it weakness at not measuring the variability in internal randomness or the choice of data set. So the test is useful if the sources of variability are small.
For getting an idea, how the difference of two proportions is, a statistical test about the difference of the error rate of two algorithms can be applied. Let p_A and p_B be the probability of incorrectly classified test samples of classifier A and B. The binomial random variable $n * p_A$ with variance $p_A * (1 - p_A) * n$ describes the number of misclassifications on n test samples. Calculating the difference between two independent normally distributed random variables is again normally distributed. If p_A and p_B are independent, the null hypothesis will be described by a mean of zero and a standard deviation error se Equation 1.46 by and the statistic z Equation 1.47:

$$se = \sqrt{\frac{2p * (1 - \frac{p_A + p_B}{2})}{n}} \tag{1.46}$$

$$z = \frac{p_A - p_B}{\sqrt{\frac{2p*(1-p)}{n}}} \tag{1.47}$$

where n is the number of test samples and z is standard normal distributed. If $|z| > Z_{0.975} = 1.96$, the null hypothesis is rejected and the two algorithms perform differently. Problems with this statistic are that the algorithms are tested on the same test set and so are not independent. Also the problem exists of not taking in account the variation due to the choice of training set or internal variations.
One of the most popular metrics is the resampled paired t test. At 30 trials, each time the test randomly divide the sample S into a training set R (normally two thirds of S) and a test set T. If for every trial the error rate of the algoritms are independent of each and normally distributed, the Student´s test can be applied by caculating the statistic:

$$t = \frac{\overline{p} * \sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n}(p^{(i)} - \overline{p})^2}{n-1}}} \tag{1.48}$$

where $\overline{p} = \frac{1}{n} * \sum_{i=1}^{n} p^{(i)}$, (i) the i-th trial and $p^{(i)} = p_A^{(i)} - p_B^{(i)}$. With n-1 degrees of freedom in a t distribution, the null hypothesis is rejected by $|t| > t_{29,0.975} = 2.045$. Here the problems are that $p_A^{(i)} and p_B^{(i)}$ are not independent so $p^{(i)}$ is not normally distributed and the $p^{(i)}$ itself is not independent, because R and T overlap in the trials.

The re sampled paired t test is called k-fold cross-validated paired t test if S is divided randomly into k sets T_k of equal size. Then in k trials, the test set is T_i and the training set is the sum of the other T_j with $j \neq i$.

1.3 Regression Metrics

In regression metrics the performance measures can be grouped in different ways. [5] recommend the division into four groups:

- primary metrics
- extended metrics
- composite metrics
- hybrid sets of metrics

Primary metrics is the mostly used group with values like Mean Absolute Error (MAE), Mean Square Error (MSE), symmetric Mean Absolute Percentage Error (sMAPE), etc... Usually they are structured in three steps: first calculating the point distance, then performing the normalization. These two steps are done for all used datapoints of the dataset. The third step is the aggregation of the calculated results of the two steps before. Often primary metrics are used for the recalculation for other metrics.
Extended metrics add some special normalization methods to the primary metrics. Here the normalization is applied after the aggregation. For example the Normalized Root Mean Square Error (NRMSE) is the Root Mean Square Error normalized by the standard deviation. The normalization can also be performed by the mean of the difference of maximum and minimum of the data.
Composite metrics combine two or more single metrics, so a single result is produced. Here a example is the Relative Root Mean Squared Error (RelRMSE), where the RMSE is divided by a RMSE from a benchmark method.
Hybrid sets of metrics use also two or more metrics for the same experiment, but they are not combined in a single mathematical structure. The idea is, that sets of metrics are developed for different purposes, where single metric complements the others for a better statement. Here [5] mention bias and accuracy as example.
[5] discuss the most common group, primary metrics, in detail. In general, primary performance measuring can be explained by:

$$m = G^{Z}_{j=1,n}[N^{Z}(D^{Z}_{(A_j,P_j)})] \tag{1.49}$$

where D is the method of point distance determining, N the method of normalization, G the method of aggregation the point distance of the dataset, A_j the actual value, P_j the predicted value, n the size of dataset and z the index of the used method. The different ways of calculating the point distance D are subtraction, with magnitude error $A_j - P_j$, absolute error $|A_j - P_j|$ and squared error $(A_j - P_j)^2$. Point distances by division, with quotient error $q_j = P_j/A_j$, absolute quotient error $|q_j|$ and squared quotient error $|q_j^2|$. Quotient error can be also used as a logatihmic quotient. Multiplication point distance is used for vector and binary data. Some examples are inner product distance (IPD), harmoinc mean distands (HMD), etc. Magnitude of error is a computationally efficient and easy method of calculation. The issue is the aggregation of the error points, which can lead to canceling by positive and negative values. The absolute error can prohibit this problem. The problem is skewness (bias) can not be calculated. The squared quotient error eliminates the problem

of the magnitude of error. The main problem is that large errors are emphasized and getting more impact on the performance metric. The logarithmic quotient error $ln(P_j/A_j)$ has the advantage of a symmetric error and is dimensionless.
Normalization brings the benefit of the comparability of different metrics with various dimensions.[5] The unitary normalization divides the metric by one and does not need further calculatoins. The normalization by actuals A_j^{-c} means the division by the actual value. Here c=1 and c=2 is possible. This metric is useless for actuals near zero. The normalization by variability of actuals $(A_j - \overline{A})^{-c}$ is calculated by division by the difference of the actual value and the mean value of all actuals. Here c=1 or c=2 is possible. It minimzes the problem of the division by actuals near zero. Instead of the mean it is possible to use the sum of actuals and predicted values $(A_j + P_j)^{-c}$. A modification is to use the average of this value by division by 2. Additionally one can use the maximum (or minimum) value of the actuals or the prediction $[max(A_j, P_j)]^{-c}$, where c=1. This normalization is used for compressed domain image retrieval. Other possibilities are the normalization by standard deviation or the difference of the actual and the predicted values.[5]

For the aggregation of primary metrics [5] recommend four methods. The (arithmetic) mean aggregation uses the normalized point distances and sums them up by division of the number of point distances. Median aggregation uses the value in the middle of the listed point distances. If the data set is even it is the mean of the two middle values. The method is more resistant against outliers. Geometric mean aggregation is the n-th root of the product of all point distances. It is also more robust with respect to outliers. One of the simplest calculation is the sum aggregation, where all D are summed. Some unusual used aggregation are the harmonic mean as the reciprocal of the arithmetic mean. The truncated mean cuts out some extreme outliers before forming the arithmetic mean. Using the Windsorized mean the extreme D are replaced by the next smallest/largest value. M-estimator weighs the D by the distance from the center of the distribution. Table 1.5 shows the possible combinations of the aggregation G, normalization N and point distance D and the shortcuts of the resulting metrics. The table with the full name of the shortcuts can be seen in the appendix.

Point Distance D	Normalizaion N					Aggregation D
	N$^1 = 1$	N$^2 = A_j^{-c}$	N$^3 =$ $(A_j - A)^{-c}$	N$^4 = (A_j + P_j)^{-c}$	N$^5 =$ $[max(A_j, P_j)]^{-c}$	
Error (magnituted error) D$^1 = A_j - P_j$	ME (MBE, bias)	MNB for c=1 MPE=100*MNB		FB c=1		G^1*Mean*
						G^2*Median*
						G^3*Geometric* Mean
	MD					G^4*Sum*
Absolute error D$^2 = \lvert A_j - P_j \rvert$	MAE (MAD)	MARE c=1 NAPE=100*MARE	MRAE c=1	FAE c=1 sMAPE=100*FAE		G^1*Mean*
	MdAE	MdAPE c=1	MdRAE c=1	sMdAPE c=1		G^2*Median*
	GMAE		GMRAE c=1			G^3*Geometric* Mean
	SAD		RAE c=1	CM c=1	WHD c=1 max	G^4*Sum*
Squared error D$^3 = (A_j - P_j)^2$	MSE RMSE = $\sqrt{MSE}$	MSPE c=2 RMSPE = $\sqrt{MSPE}$				G^1*Mean*
		MdSPE c=2 RMdSPE=$\sqrt{MdSPE}$				G^2*Median*
	GRMSE					G^3*Geometric* Mean
	SSE ED = $\sqrt{SSE}$	NCSD c=1	RSE c=2 RRSE=$\sqrt{RSE}$	SquD c=1 DivD c=2	VSD c=1 min	G4 Sum
Log quotient error D$^4 = ln(P/A)$						G^1*Mean*
	MdLAR					G^2*Median*
						G^3*Geometric* Mean
		KLD c=1				G^4*Sum*
Absolute Log quotient error D$^5 = \lvert ln(P/A) \rvert$	MNAFE					G^1*Mean*
	MdSA					G^2*Median*
						G^3*Geometric* Mean
						G^4*Sum*

Table 1.5: Regression Metrics for different point distances (D), normalizations (N) and aggregations (D). The explanation of the shortcuts can be found in the appendix (Source: [5])

1.4 Software Python

Python is the preferred software for implementing the LM. It is a object-oriented, open source, free available coding language, especially common in science for interpreting large amount of data, solving mathematical problems and building different kinds of neuronal networks. For these applications Python has different open source libraries. In this thesis the used library for ML is scikit-learn, short sklearn. The library include the most important ML algorithms and corresponding metrics. For the preparing of the data pandas is the used library. In pandas dataframe objects simplify the handling of the data. [4] [3] [20].

The package fitter [2] is a package for fitting the distribution of data by a the probability distributions of the package SciPy [23]. Table 1.6 shows all the used distributions

alpha
anglit
arcsine
argus
beta
betaprime
bradford
burr
burr12
cauchy
chi
chi2
cosine
crystalball
dgamma
dweibull
erlang
expon
exponnorm
exponweib
exponpow
f
fatiguelife
fisk
foldcauchy
foldnorm
genlogistic
gennorm
genpareto
genexpon
genextreme
gausshyper
gamma
gengamma
genhalflogistic
genhyperbolic
geninvgauss
gibrat
gompertz
gumbel_r
gumbel_l
halfcauchy
halflogistic
halfnorm
halfgennorm
hypsecant
invgamma
invgauss
invweibull
johnsonsb
johnsonsu
kappa4
kappa3
ksone
kstwo
kstwobign
laplace
laplace_asymmetric
levy
levy_l
levy_stable
logistic
loggamma
loglaplace
lognorm
loguniform
lomax
maxwell
mielke
moyal
nakagami
ncx2
ncf
nct
norm
norminvgauss
pareto
pearson3
powerlaw
powerlognorm
powernorm
rdist
rayleigh
rice
recipinvgauss
semicircular
skewcauchy
skewnorm
studentized_range
t
trapezoid
triang
truncexpon
truncnorm
truncweibull_min
tukeylambda
uniform
vonmises
vonmises_line
wald
weibull_min
weibull_max
wrapcauchy
multivariate_normal
matrix_normal
dirichlet
wishart
invwishart
multinomial
special_ortho_group
ortho_group
unitary_group
random_correlation
multivariate_t
multivariate_hypergeom
bernoulli
betabinom
binom
boltzmann
dlaplace
geom
hypergeom
logser
nbinom
nchypergeom_fisher
nchypergeom_wallenius
nhypergeom
planck
poisson
randint
skellam
yulesimon
zipf
zipfian

Table 1.6: distributions the package fitter uses from SciPy for fitting the data (source: [23])

1.5 Kolmogorow-Smirnow-Test

The Kolmogotow-Smirnow (KS) test is a statistical computation to check if a variable belongs to a probability distribution. The zero-hypothesis H_0 of the test is shown in Equation 1.50. The hypothesis show that the variable X can be described by the probability distribution F_0.

$$H_0 : F_X(x) = F_0(x) \tag{1.50}$$

The alternative hypothesis H_1 describe, that the variable X can be described by another probability distribution F_0 [14]. Figure 1.4 shows the distributions for the accuracy calculation of a KNN predictor. The predictor was trained and tested 300 times. Table 1.7 lists the p-values for the plotted distributions. Here the t-distribution shows the highest p-value. For the given confidence interval, all distributions with a p-value>0.05 can describe the data. For them the H_0 is right.

Distribution	***p-value***
t	0.560755
kappa4	0.558511
johnsonsu	0.488186
dweibull	0.368915
dgamma	0.293223

Table 1.7: p-values for the distributions shown in Figure 1.4

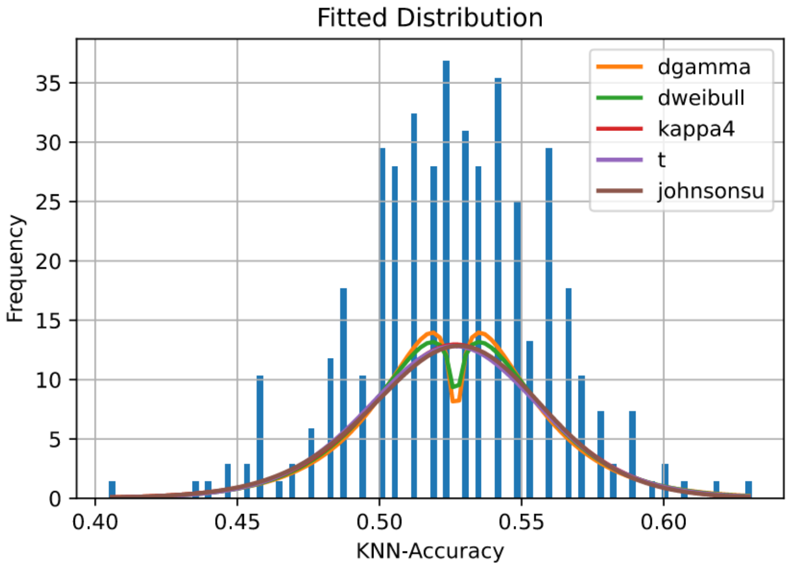


Figure 1.4: Histogram for the accuracy of KNN predictor

2 Data and Machine Learning Algorithm

The link to the Github server can be found here: https://github.com/rmttugraz/MatthiasHann.git. All used data sets, codes and the histograms can bed found here. Later in this thesis, there will be references to the data on this server.

2.1 Data preparation Classifiaction

The data for training the LM are chosen from the homepage of the "TC304 Engineering Practice of Risk Assessment & Management" of the International Society of Soil Mechanics and Geotechnical Engineering (ISSMGE). For classification 7 data sets are used. The data set C#1 [17] contains coarse grained soils, clays from Finnland, from 176 studies. The input and the classes for the LM are described in the Listing 2.1. All classes with less than 20 samples are removed from the data set. So 6 classes are given, as described below.

```
X = data[["Void ratio", "Water content (%)", "Unit weight (kN/m3)", "Effective in-situ stress (kPa)", "Pre-consolidation pressure (kPa)", "OCR", "Compression index", "Swelling index"]]
y = data['Soil typea'].values

Soil typea
Sa             241
liSa           158
laSa            63
Sa/Si           45
Sa/Lj           29
ljSa            21
```

Listing 2.1: Finish Clays; data set C#1

Data set C#2 [13] describes the compressive strength, tensile strength and friction properties for pyroclasts, andesits, basalts and dacite. Listing 2.2 shows the input data X and the classes y.

```
X = data[['porosity','UCS (MPa)', 'Youngs modulus (GPa)']]
y = data['rock_type_cat'].values
```

Listing 2.2: Volcanic Rocks; data set C#2

The C#3 data set [9] contains deformation modulus, elastic modulus, dynamic modulus, rock quality designation, rock mass rating, Q-system, geological strength index of a rock mass as well as intact-rock Young's modulus and intact-rock uniaxial compressive strength for 5876 rock mass cases. Listing 2.3 shows the input data X and the classes y. The classes are more or less balanced, beside the mudstone.

```
X = data[['Sigma ci (MPa) (intact rock)', 'RQD']]
y = data['Rock Type'].values

Rock Type
mudstone         104
gneiss            66
sandstone         62
andesite          35
basalt            31
siltstone         31
```

```
Serpentinite      30
limestone         29
Syenite           27
granite           27
```

Listing 2.3: Rock Masses of different kind of rocks; data set C#3

Data set C#4 [6] contains of 27.5% igneous rock, 59.4% sedimentary rocks and 13.1% metamorphic rock. Unit weight (kN/m^3), Dry unit weight (kN/m^3), water content (%), Porosity (%), Effective Porosity (%), Schmidt hammer hardness (RL), Shore scleroscope hardness(Sh), Is50 (MPa), P-wave velocity(km/s), σ_t Brazilian(MPa), UCS(MPa), Young's modulus E (GPa). Listing 2.5 shows the input data X and the classes y. The classes are more or less balanced, beside the limestone.

```
X = data[['Dry unit weight (kN/m^3)', 'P-wave velocity(km/s)', 'UCS(MPa)']]
y = data['Rock Type'].values

Rock Type
Limestone     75
Granite       55
Serpentine    52
Peridotite    35
```

Listing 2.4: Mixed Stone; data set C#4

[24] describes data set C#5 and contains shear wave velocity and the blow count (N) from the SPT. Listing 2.5 shows the input data X and the classes y. The classes are large enough so data imbalance is not so important, beside Silty soils.

```
X = data[["N", "Vs (m/s)"]]
y = data['Soil type'].values

Soiltype
Sandy soils                849
Clayey soils               441
Sandy silt/silty sand      260
Silty soils                229
Soft to stiff soil         180
Silty soils                 35
```

Listing 2.5: N-Vs correlation; data set C#5

The data set C#6 is downloaded from the homepage of the TU Graz, (Source: [1]). It is a data set from the Geotechnik Premstaller GmbH from hundrets of cone penetration tests. Listing 2.6 shows the input data X and the classes y. The data are highly imbalanced.

```
X = data[['Depth (m)','qc (MPa)','fs (kPa)','Vs (m/s)']]
y = data["Oberhollenzer_classes"].values

Oberhollenzer_classes
7.0                        616
6.0                        476
1.0                        337
5.0                        304
0.0                         92
4.0                         54
2.0                         47
3.0                         24
```

Listing 2.6: CPT Getoechnik Premstaller; data set C#6

The data set C#7 [12] describe the properties of saturated hydraulic conductivity and Atterberg Classification for fine grained soils. Listing 2.7 shows the input data X and the classes y. The data set is highly imbalanced.

```
X = data[["void ratio", "k (m/s)", "wL", "wP", "IP", "Gs"]]
y = data[’Atterberg classification’].values

Atterberg classification
CH                            366
CL                            230
MH                            184
ML                             59
```

Listing 2.7: Fine grained soils (clay); data set C#7

2.2 ML Algorithm Classifier

After deciding which input data should be classified the ML algorithm are chosen. Listing 2.8 shows the 10 used algorithms. All are loaded from the package scikit-learn and the knowledge of the description of them in the following part is taken from [20].

```
RANDOM_SEED=42

rfc = RandomForestClassifier(random_state=RANDOM_SEED)
knn = KNeighborsClassifier()
svm = SVC(random_state=RANDOM_SEED, probability=True)
dtc = DecisionTreeClassifier()
gnb = GaussianNB()
lda = LinearDiscriminantAnalysis()
abc = AdaBoostClassifier(random_state=RANDOM_SEED)
qda = QuadraticDiscriminantAnalysis()
mlp = MLPClassifier(max_iter = 4000, random_state=RANDOM_SEED)
lrc = LogisticRegression(solver=’sag’, max_iter=4000, random_state=
    RANDOM_SEED)
```

Listing 2.8: Classifier algorithm

The Random Forest Classifier work on three main steps. After selecting random samples from a data set, the algorithm constructs a decision tree for each sample. There it predicts a result for every tree. Every predicted result gets a vote/weight in comparison with the test set. The best voted prediction is the final prediction. So the decision tree is trained for regression and classifying tasks. In general RFC are robust because of the high number of decision trees. Overfitting is only a small problem because of averaging the predictions. The architecture of the algorithm causes high computational cost because every decision tree gives a prediction and after the prediction must be voted. In summary a Random Forest Classifier/Regressor is a set of multiple decision trees.

The K-Neighbors Classifier is based on the nearest neighbors classification. Here the algorithm does not construct a model for general prediction. The algorithm stores instances of the training data set and so uses them for the classification by a simple majority vote of the nearest neighbor. The class will be assigned by a query point. This point is the most representative of the class. Weights are used to give the neighbors different impacts on the prediction. Figure 2.1 shows a 3-Class KNN Classifier with uniform weights and the number of neighbors per class k=15.

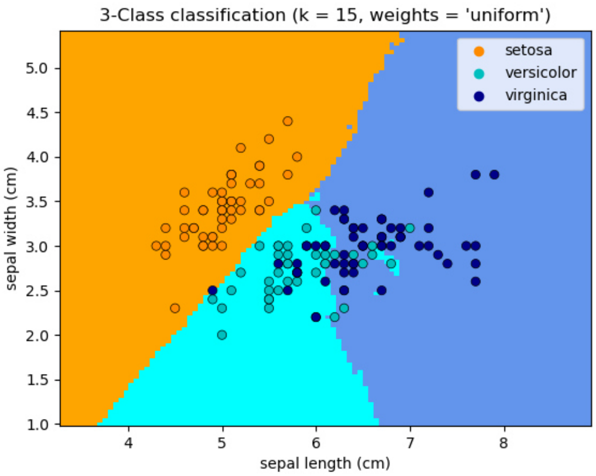


Figure 2.1: An example of a 3-Class KNN-Calssifier. (Source: [20])

Support Vector Machines (SVM) divide the input data into a high-dimensional feature space and classify them by this space. Depending on the form of hyperplane (linear, polynomial, radial, sigmoid,...), which divides the feature space, the complexity of the classification changes. The decision for the best hyperplane is taken by measuring the maximum margin hyperplane-nearest data point. SVM are very effective for high dimensional spaces and for small sample numbers. If the feature space is much higher than the number of samples it is import to choose the correct kernel function (hyperplane) to avoid over-fitting.

In comparison with the Random Forest Classifier, the Decision Tree Classifier uses only one decision tree instead of a collection of decision trees. This makes the algorithm computational less expensive, but also less stable. Figure 2.2 shows an example of a decision tree, trained by the data set #1 described before. One can see how complex the structure is.

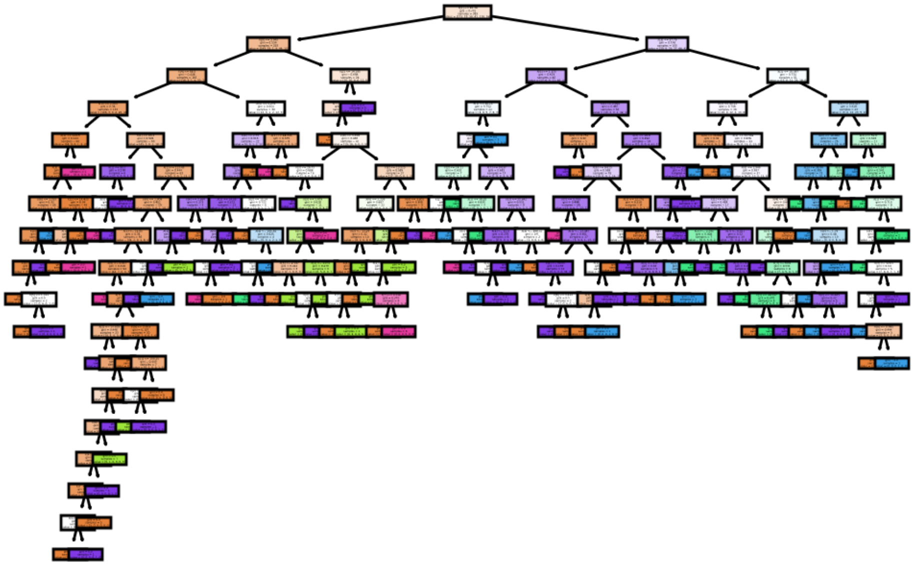


Figure 2.2: Decision Tree trained by the data set #1

The Gaussian Naive Bayes (GNB) use the Equation 2.1 for classifying the data. It is based on the Gaussian distribution. The assumption is that the features are independent of each other. For every class a Gaussian distribution with standard deviation is given. The algorithm classifies by calculating the probability that the data point belongs to the class. Equation 2.1 shows the implemented equation for the GNB. σ_y and μ_y are chosen by the maximum likelihood method.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(\frac{-(x_i-\mu_y)^2}{2\sigma_y^2}\right) \tag{2.1}$$

Linear Discriminant Analysis, LDA, describes a linear decision surface, which is used to separate the classes. The Quadratic Discriminant Analysis, QDA, uses a quadratic surface. The linear method limits the complexity of the class in comparison to the quadratic. The Bayes´rule is used, modified $P_{(x|y)}$ as a multivariate Gaussian distribution with the density function as shown in Equation 2.2, where d is the number of features and k the class.

$$P(x|y=k) = \frac{1}{(2\pi)^{d/2}|\sum_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^t \sum_k^{-1}(x-\mu_k)\right) \tag{2.2}$$

The QDA is described by the log of the above equation, seen here Equation 2.3. C_{st} describes the constant terms of the Gaussian. This equation gets maximum by the class with the best prediction.

$$logP(x|y=k) = logP(x|y=k) + logP(y=k) + C_{st} = -\frac{1}{2}log|\sum_k| - \frac{1}{2}(x-\mu_k)^t \sum_k^{-1}(x-\mu_k) + logP(y=k) + C_{st} \tag{2.3}$$

For the LDA, a variation of the QDA, it is assumed that all classes share the same covariance matrix. One can describe the above equation by Equation 2.4, where $\omega_k^t x = \Sigma^{-1} \mu_k$ and $\omega_{k0} = -\frac{1}{2}\mu_k^t \Sigma^{-1} \mu_k + logP(y = k)$ describes.

$$logP(x|y = k) = \omega_k^t x + \omega_{k0} + C_{st} \tag{2.4}$$

AdaBoostClassifier uses the AdaBoost algorithm. The principle is to train weak learners, like small decision trees, repeated on modified versions of the data. The final predictor is formed by a weighted majority vote of all these predictors. Each boosting iteration consists of applying weights to each training sample. The weights are increased for the incorrectly performing predictors and vice versa for the correctly performing predictors. So each subsequent weak learner is forced on the missed samples. The error rate decreases with increasing of number of weak learners. Figure 2.3 shows correlation between the error rate, the number of weak learners and the chosen predictor.

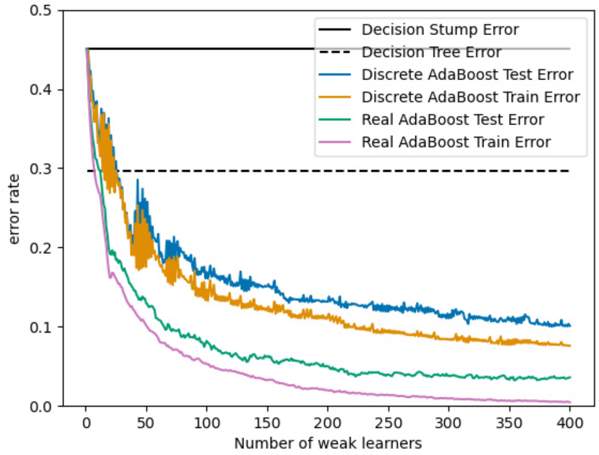


Figure 2.3: Correlation between number of weak learners in the AdaBoost method, error rate and the chosen predictor (source: [20]

The Multi-Layer Perceptron, MLP, belongs to the group of deep learning algorithm. The neuronal network is built up by an input and an output layer and one or more hidden layers in between. The layers are built by neurons. A neuron calculates from all inputs a weighted sum, like $x_1w_1 + x_2w_2 + ... + x_nw_n$. This sum is the input for a following non-linear activation function, like a sigmoidal function. The outputs of the activation function of all neurons of a layer is the input for every neuron in the next layer. The bias is a constant which is added to the neuron summation to give the output a minimum value. MLP is able to learn non-linear-models and real-time/online. The loss function of the hidden layers contains more than one local minimum, so with randomly initiated weights the MLP can lead to different accuracies. Choosing the architecture parameters like number of hidden neurons and layers and number of iterations is a trail and error process or needs experience. The scaling of the features affect the MLP. In the case of classification, the MLP is trained by backpropagation. First the network is built up and classifies one time the inputs with the initialized weights. Second the failure by the supervised data set is calculated. Third the failure value is back calculated from the output layer until the input layer. The weights in the neurons are changed depending on their input on the failure. The fourth step, the prediction is repeated like in step two. The iteration follows the minimum of the failure or until a maximum number of iteration is reached. Figure 2.4 shows the basic architecture of a neuronal network.

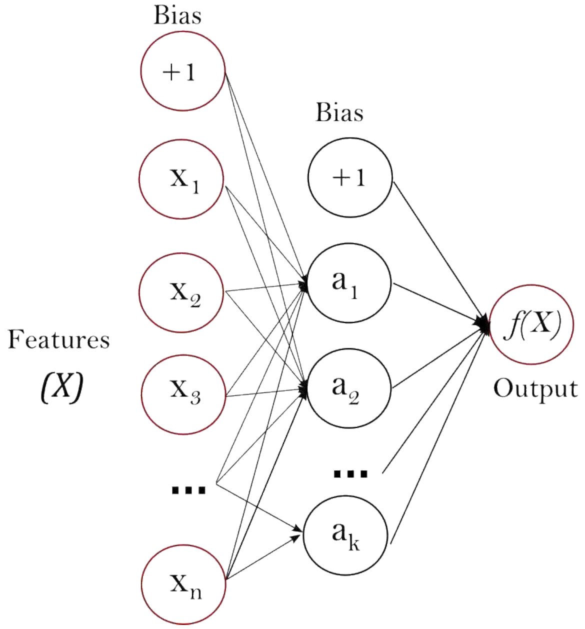


Figure 2.4: A neuronal network with input layer, hidden layer and output layer. a describe the neurons. (source: [20]

Logistic-Regression LR is a part of linear models, in which the supervisor assumes that the target can be described by a linear combination of the features $y_{(w,x)} = w_0 + w_1 x_1 + ... + w_p x_p$. LR is used for classification rather than in regression and uses a logistic function Equation 2.5 for modeling the probabilities describing the possible outcomes.

$$f_{(c)} = \frac{L}{1 + e^{-k(x-x_0)}} \tag{2.5}$$

2.3 Data preparation Regression

Again all data for training the LM are chosen from the homepage of the "TC304 Engineering Practice of Risk Assessment & Management" of the International Society of Soil Mechanics and Geotechnical Engineering (ISSMGE). For regression 9 data sets are available. Scatter plots are used to check which input data X correlate with the output y. Figure 2.5 shows such a scatter-plot. Here the correlation is shown between vertical preconsolidation pressure and undrained shear strength, from data set R#1 which is described later.

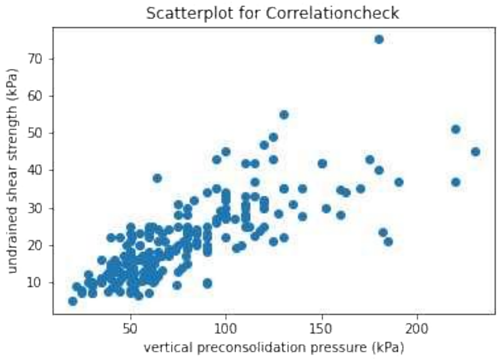


Figure 2.5: An example for checking if an input feature X shows correlation with predicted output y

Data set R#1 [11] describes the correlation of undrained finish clays. Field and laboratory measurements from 24 test sites are given. The input features X are effective vertical stress (s'v), vertical preconsolidation pressure (s'p), natural water content (w), liquid limit (LL), plastic limit (PL) and sensitivity (St=s_u/s_u^{re}). The predicted value is s_u from field vane test (s_u^{FV}). Listing 2.9 shows X and y.

```
X = data[["LL(%)", "PL(%)", "w(%)", "s'v (kPa)", "s'p (kPa)", "St"]]
y = data["su(test) (kPa)"]
```

Listing 2.9: Correlation for undrained finish clays; data set R#1

R#2 [1] is the same data set as C#6 are equal. The difference is that the shear wave velocity is now the value target instead of an input variable. Listing 2.10 shows the code.

```
X = data[['Depth (m)','qc (MPa)','fs (kPa)']]
y = data['Vs (m/s)']
```

Listing 2.10: CPT Getoechnik Premstaller; data set C#6

R#3 [6] is the same data set as C#4. Listing 2.11 shows the input feature X and the predicted output y for the given data set.

```
X = data[["Porosity (%)","Schmidt hammer hardness (RL)","Is50 (MPa)","P-wave velocity(km/s)"]]
y = data['UCS(MPa)']
```

Listing 2.11: Mixed Stone; data set R#3

R#4 [8] normalized undrained shear strength ($su/\sigma_v^{'}$, where $\sigma_v^{'}$ is the vertical effective stress), overconsolidation ratio ($OCR = \sigma_p^{'}/\sigma_v^{'}$, where $\sigma_p^{'}$ is the preconsolidation stress), normalized cone tip resistance $(q_t - \sigma_v)/\sigma_v$ (where σ_v is the vertical total stress), normalized effective cone tip resistance $(q_t - u_2)/\sigma_v^{'}$, normalized excess pore pressure $(u2 - u0)/\sigma_v^{'}$ (where u_0 is the static pore pressure), and pore pressure ratio $B_q = (u_2 - u_0)/(q_t - \sigma_v)$. Listing 2.12 shows the input feature X and the predicted output for the given data set.

```
X = data[["su/σ'v", "OCR", "(qt - σv)/σ'v", "(qt  -u2)/σ'v", "(u2 - u0)/σ'v", "Bq"]]
y = data["σ'v(kPa)"]
```

Listing 2.12: Mixed Stone; data set R#4

R#5 [16] contains data from 124 test sites in the Jiangsu province. The data set is measured by piezocone testing devices and contains of resilient modulus (Mr) values at the in-situ stress condition, cone tip resistance (qc), sleeve frictional resistance (fs), moisture (w) and dry density (γd). Listing 2.13 shows the input feature X and the predicted output y for the given data set.

```
X = data[['qc (MPa)', 'fs (MPa)', 'w (%)', 'γd (kN/m\xa03)']]
y = data["Mr (MPa)"]
```

Listing 2.13: Jiangsu province piezocone testing indices; data set R#5

R#6 [26] is a combination for 11 clay parameters covering 50 sites in Shanghai with 4051 data points, covering an area of 145 km^2. Eleven parameters are given: LL(%), PI, LI, e, K0, $\sigma_v^{'}$ (kPa), Su(UCST) (kPa), St(UCST), Su(VST) (kPa), St(VST), ps (MPa), $\sigma_v^{'}$ (Pa), Su(UCST)/$\sigma_v^{'}$, Su(VST)/$\sigma_v^{'}$, ps/$\sigma_v^{'}$. Listing 2.14 shows the input feature X and the predicted output y for the given data set.

```
X = data[['PI',"ps/σ'v","e","ps\n(MPa)"]]
y = data['LL(%)']
```

Listing 2.14: Shangha clays; data set R#6

R#7 [13] is the same data set as C#2. Listing 2.14 shows the input feature X and the predicted output y for the given data set.

```
X = data[['porosity','Youngs modulus (GPa)']]
y = data['UCS (MPa)']
```

Listing 2.15: Volcanic rocks; data set R#7

R#8 [7] contains data of coarse grained soils from 176 studies. The data set provide the parameters Fines(%), D50(mm), Cu, Dr(%), OCR $\sigma_v^{'}$(kPa), (N1)60, qt1, $\phi_{cv}()$ and $\phi_p()$. Listing 2.16 shows the input feature X and the predicted output y for the given data set.

```
X = data[["σv'(kPa)", "(N1)60", "qt1", 'Dr(%)']]
y = data['ϕp(◦)']
```

Listing 2.16: Data set of coarse-grained soils; data set R#8

Data set R#9 [18] describes the correlation of soft finish clays. The data are based on the oedometer test. The input features X are water content,fineness number, undrained shear strength, pre-consolidation pressure, OCR, compression index and the swelling index. The predictor output is the void ratio. Listing 2.17 shows X and y.

```
X = data[['Water content (%)','Fineness number','Undrained shear stregth (
    fall cone) (kPa)', 'Pre-consolidation pressure (kPa)', 'OCR', '
    Compression index', 'Swelling index']]
y = data['Void ratio']
```

Listing 2.17: Data set of soft finish clays; data set R#9

2.4 ML Algorithm Regression

For regression, 10 algorithms are chosen. Listing 2.18 shows them. All are loaded from the package scikit-learn and the knowledge of the description of them in the following part is taken from [20].

```
lir = LinearRegression()
las = Lasso(random_state=RANDOM_SEED)
svr = svm.SVR()
```

```
rid = Ridge(random_state=RANDOM_SEED)
enc = ElasticNetCV(precompute='auto', random_state=RANDOM_SEED)
mlp = MLPRegressor(max_iter = 2000, random_state=RANDOM_SEED)
dtr = DecisionTreeRegressor(random_state=RANDOM_SEED)
rfr = RandomForestRegressor(random_state=RANDOM_SEED)
knn = KNeighborsRegressor()
gpr = GaussianProcessRegressor(random_state=RANDOM_SEED)
```

Listing 2.18: Regression algorithm

Here the Linear Regression LR is based on the ordinary least squares method. The algorithm fits a linear model with coefficients ω with the aim to minimize the residual sum of squares between predicted and supervised output. Equation 2.6 describes the mathematical basics behind the least square method and Figure 2.6 the graphical interpretation of a fitted data set. $X\omega$ is the weighted input which predicts the output and y is the observed output from the data set.

$$min_{\omega}||X\omega - y||_2^2 \tag{2.6}$$

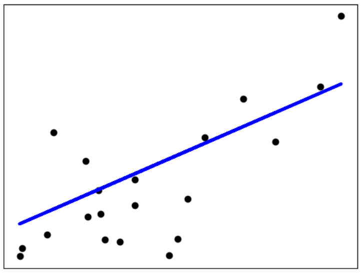


Figure 2.6: The ordinary least square method in the sklearn package (source:[20])

The Lasso method belongs also to the group of linear models. It needs fewer coefficients in comparison to other linear methods. The method is used in compressed sensing, where the underdetermining of linear systems is common. Mathematically, the Lasso method can be described like in Equation 2.7. Here the term α is a constant and $||\omega||_1$ is the l_1-norm of the coefficient vector.

$$min_{\omega}\frac{1}{2n_{samples}}||X\omega - y||_2^2 + \alpha||\omega||_1 \tag{2.7}$$

Like in the Support Vector Classification SVC, Support Vector Regression SVR depends on a subset of the training data. Here the cost function does not take in account the points that lie beyond the margin. In the basics, the SVR and the SVC work in the same way.

Another linear model is the Ridge Regression. Like the Lasso method, the ordinary square method is extended by the same extra term, but here the squared l_2-norm of the coefficient vector is used, like in Equation 2.8. Figure 2.7 shows the dependence of the constant α and the weights. By increasing α the coefficients become more robust to collinearity

$$min_{\omega}\frac{1}{2n_{samples}}||X\omega - y||_2^2 + \alpha||\omega||_2^2 \tag{2.8}$$

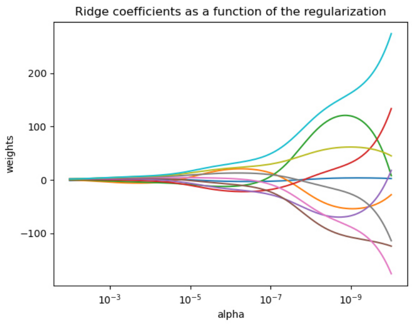


Figure 2.7: Robustness of collinearity of the coefficients in dependence of α (source:[20])

The Elastic-Net depends on the linear models and is based on l_1- and l_2-norm regularization of the coefficients. It combines the advantages of Lasso and Ridge. If the feature vector contains multiple related features, the Elastic-Net is very useful. The function for minimizing is shown in Equation 2.9. Figure 2.8 shows the difference of the impact of α on the coefficients between Lasso and Elastic-Net.

$$min_{\omega} \frac{1}{2n_{samples}} ||X\omega - y||_2^2 + \alpha\rho||\omega||_1 + \frac{\alpha(1-\rho}{2}||\omega||_2^2 \tag{2.9}$$

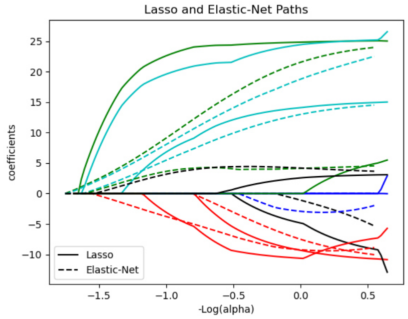


Figure 2.8: Combaring influence of α on coefficients for Lasso and Elastic-Net (source:[20])

The difference between the MLP Classifier and MLP Regressor is, that the MLP Regressor uses no activation function in the output layers. So continuous values are generated as output.

Decision Tree Regression DTR works in the same way as described above for DTC. DTR predict a continuous output.

Also the Random Forest Regressor works like the Random Forest Classifier, described above.

KNN Regressor works like the KNN Classifier, described above.

The Gaussian Processes GP is a probabilistic (Gaussian) method, so for the decision of refitting the predictor in some regions, the confidence intervals can be computed. Interpolation happens by the prediction of the observation. The problem is, when the number of features becomes more than a few dozens, the algorithm loses efficiency. The Regressor implements the GP for regression.

2.5 Used Metrics

Listing 2.19 lists the used classification metrics. In sklearn, the accuracy is calculated like described above, as the sum of the correct predicted samples averaged by the number of samples.

```
acc = accuracy_score()
f1 = f1_score()
fb = fbeta_score()
hamming = hamming_loss()
jaccard = jaccard_score()
log = log_loss()
prec = precision_score()
rec = recall_score()
zero = zero_one_loss()
```

Listing 2.19: Used Classification Metrcis

F_1 and F_β are a harmonic mean, computed from the recall and the precision. Equation 2.10 describes the F_β score. If $\beta = 1$, F_1 and F_β are equal.

$$F_\beta = \frac{(1+\beta^2)*TP}{(1+\beta^2)*TP+\beta^2*FN+FP} \tag{2.10}$$

The Hamming loss describes the sum of the wrong labeled values divided by the number of labels. Jaccard similarity coefficient, known as Jaccard index, is the average of the Jaccard similarity between paired label sets. The Jaccard similartiy is described in Equation 1.3.

$$J_{(y_i,\hat{y}_i)} = \frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|} \tag{2.11}$$

Log loss is a metric based on probability estimations. It calculates instead of discrete outputs the probability predictions. With the probability $p = Pr(y = 1)$, the negative log-likelihood of the TP is the log loss. For a binary classifier Equation 2.12 and for multi class predictor Equation 2.13 is used.

$$L_{log}(y,p) = -logPr(y|p) = -(ylog(p)+(1-y)log(1-p)) \tag{2.12}$$

$$L_{log}(Y,P) = -logPr(Y|P) = -\frac{1}{N}\sum_{i=0}^{N-1}\sum_{k=0}^{K-1} y_{i,k}log(p_{i,k}) \tag{2.13}$$

Precision measures the performance of not labeling a positive sample as negative and Recall measures the ability to find all positive samples by a predictor.

Zero one loss is the metric for computing the sum or average of L_{0-1} classification loss by all samples. Equation 2.14 shows how zero one loss is calculated.

$$L_{0-1}(y_i,\hat{y}_i) = 1(\hat{y}_i \neq y_i) \tag{2.14}$$

Listing 2.20 shows the used regression metrics in the project. The max error is the maximum difference between the predicted and supervised (correct) target values. The mean absolute error is the sum of all absolute values of the residual error divided by the number of output values. The mean squared error is calculated by the sum of the square of the residual error divided by the number of predictions.

```
mer = max_error()
mae = mean_absolute_error()
```

```
mse = mean_squared_error()
msle = mean_squared_log_error()
mee = median_absolute_error()
r2 = r2_score()
```

Listing 2.20: Used Regression Metrcis

The mean squared log error is calculated by Equation 2.15.

$$MSLE_{(y,\hat{y})} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (log_e(1+y_i) - log_e(1+\hat{y}_i))^2 \tag{2.15}$$

The median absolute error calculates the absolute errors and takes the median of it. R^2 score, also known as the coefficient of determination, describes the ratio of the variation of the dependent variable, predictable by the independent variable/s. It is a indicator of how well unseen samples are predicted by the model, explained by variance. At all, R^2 is not comparable across different data sets. Values from <0 and >1 are possible. In most cases, the range varies between 0 and 1. Equation Equation 2.16 shows the prediction of the R^2 value, where $\bar{y}$ is the average of all y_i.

$$R^2_{(y,\hat{y})} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.16}$$

3 Training the Algorithm

One of the major questions is, how to rate the performance of a metric in comparison to an algorithm. To give a common recommendation, the metric in context with the algorithm, should show a stable performance over different data sets. These data sets should show the same or similar inputs and the same output target. So, for example, when the algorithm is trained on the blow count of a standard penetration test and the friction of the jacket, for similar soil conditions, the performance of the algorithm should be similar for different data sets. Therefore dozens or hundreds of data sets would be needed to give a statistically relevant recommendation. The problem is, such amount of data sets are not available for this thesis. At all, 16 data sets for classification and regression are provided from the ISSMGE for supervised machine learning. Another problem is the subjectivity of the choosen parameters for the algorithm and metric. For example, the MLP algorithm provides different selectable parameters like the activation function or the solver. Here one can select between identity, logistic, tanh and relu activation function or between lbfgs, sgd and adam solver. To solve these problem, the choosen algorithms are trained on the same data sets several times. At every loop the data set is randomly split into a training and testing part and so the performance can be measured. After doing this loop several times, the calculated metrics should show a probability distribution with a small standard deviation. The data sets, described above, with the selected features, algorithm and metrics are trained and measured 300 times.

Figure 3.1 shows the distribution of the metric F1-score for the KNN predictor on data set C#1.

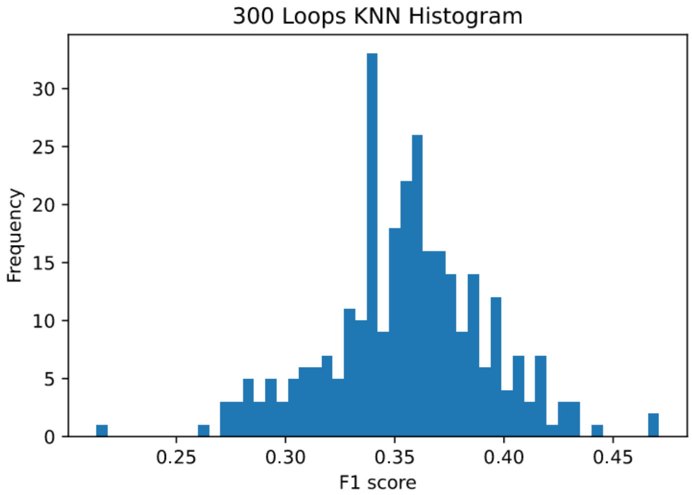


Figure 3.1: Distribution of F1-score of KNN algorithm on data set C#1.

The package Fitter fits the distribution of the calculated metric with the probability distributions of the package SciPy. Table 3.1 shows the output of the performance and statistical values of the Fitter package for the Figure 3.1. The sumsquare_error describes the squared failure between the data point and the corresponding calculated point of the fitted distribution, summed over-all points. Equation 3.1 shows the implemented formula in the pacakge Fitter [2]. The ks_pvalue describes the Kolmogorow-Smirnow-test p value.

$$\sum_{i}(Y_i - pdf(X_i))^2 \tag{3.1}$$

distbribution	***sumsquare_error***	***aic***	***bic***	***kl_div***	***ks_statistic***	***ks_pvalue***
gennorm	676.6848	-80.5612	261.1383	inf	0.031212	0.922978
hypsecant	683.1286	-91.8865	258.2778	inf	0.032623	0.896384
dweibull	690.6456	-87.6537	267.2646	inf	0.035475	0.831369
mielke	700.4297	-81.615	277.1886	inf	0.037058	0.790198
genlogistic	699.8671	-84.0702	271.2437	inf	0.037217	0.785893
burr	702.0244	-80.3394	277.8708	inf	0.037698	0.77277
dgamma	694.5576	-92.4281	268.9591	inf	0.038771	0.742806
fisk	701.2743	-78.4744	271.8463	inf	0.038961	0.737426
logistic	701.2754	-80.4753	266.143	inf	0.038965	0.73729
norminvgauss	703.786	-79.6647	278.6227	inf	0.039241	0.72945

Table 3.1: Output of the fitter package for the fitted probability distributions of Figure 3.1.

The probability distribution is chosen based on the p value. Higher p-value implements a higher probability that the H_0 hypothesis is true and the data can be described better with the investigated probability distribution than with another probability distribution. For comparability between the metrics, one distribution is chosen for all metrics of an algorithm. So it is possible to give a recommendation for one algorithm over-all metrics. sum_rows is the sum of all metrics of one fitted probability distribution. Table 3.2 is sorted by sum_rows and so the probability distribution, which fits best the data can be chosen. In this example the mielke probability distribution is ranked highest. Also all p-value for every metric of the mielke distribution is higher than 0.05. So the mielke is chosen. In some cases, the best rated probability distribution has not a p-value>0.05 for every metric. In such cases a compromise between high ranking and all p-values>0.05 is taken.

distribution	***acc***	***f1***	***fb***	***hamming***	***jaccard***	***log***	***prec***	***rec***	***zero***	***sum_rows***
mielke	0.379065	0.790198	0.960741	0.418083	0.926498	0.724588	0.987807	0.850051	0.418083	6.455115
genlogistic	0.411325	0.785893	0.956997	0.419908	0.917143	0.54195	0.991856	0.866894	0.419908	6.311874
gennorm	0.43286	0.922978	0.871308	0.432919	0.984672	0.581619	0.746924	0.783503	0.432919	6.189702
logistic	0.413052	0.73729	0.838792	0.413052	0.915697	0.648727	0.835029	0.761601	0.413052	5.976292
burr12	0.415499	0.579736	0.830621	0.502064	0.939503	0.354697	0.963183	0.782688	0.502064	5.870055

Table 3.2: Ranked distributions and metrics for the KNN algorithm of data set C#1, based on the p-value.

With the chosen probability distribution the parameters for every metric distribution are calculated. The parameters are needed to calculate the normalized standard deviation. In best case, the standard deviation tends to zero. A small standard deviation indicates a low variation of the data. For all algorithm this work flow is repeated.

4 Result of the Loop

4.1 KS p-value

The following tables show the results of the 300 time training and performance measuring loop. The red cells show p-values < 0.05. The histrograms of the p-value < 0.05 algorithm-metrics distributions are plotted in the appendex.

Table 4.1 shows the p-values and chosen distributions of data set C#1. The ABC algorithm includes 4 distributions which are not reaching the p-value > 0.05 criteria. The hamming-loss and the zero-one-loss show for multi class predictor the same value. The Figure A.1 shows, that the hamming-loss and the precision are not log-laplace probability distributed. The log-loss plot shows the form of the chosen probability distribution, so the reason for no p-value output is a problem of fitting the distribution by the fitter package.

KS p-value	***acc***	***f1***	***fb***	***hamming***	***jaccard***	***log***	***prec***	***rec***	***zero***	**sum_rows**	**distribution**
rfc	0.57997	0.708073	0.993205	0.521625	0.791577	0.485802	0.951955	0.864268	0.52163	6.418108	exponnorm
knn	0.206091	0.919687	0.825962	0.206091	0.875304	0.921724	0.95284	0.872892	0.206091	5.986682	beta
svm	0.197008	0.857426	0.441192	0.597949	0.790008	0.753941	0.224489	0.989998	0.597949	5.449961	loggamma
dtc	0.453821	0.964398	0.968934	0.432827	0.94412	0.432597	0.929211	0.975831	0.432827	6.534565	exponnorm
gnb	0.145032	0.981891	0.995995	0.241883	0.759843	0.992467	0.944814	0.903024	0.241883	6.206832	burr12
lda	0.126985	0.829815	0.859142	0.126995	0.644694	0.836764	0.575266	0.61704	0.126995	4.743695	skewnorm
abc	0.071947	0.350253	0.060927	0.01334	0.844859	NaN	0.000557	0.190476	0.01334	1.545699	loglaplace
qda	0.324811	0.954001	0.868535	0.300234	0.954092	0.542471	0.900821	0.981646	0.300234	6.126844	exponnorm
mlp	0.251229	0.989814	0.968402	0.251219	0.995221	0.944352	0.901505	0.999394	0.251219	6.552357	skewnorm
lrc	0.140093	0.851027	0.755936	0.140093	0.863189	0.927086	0.5666	0.532109	0.140093	4.916225	logistic

Table 4.1: The p-values and the chosen distribution of C#1. In red p-values < 0.05.

For binary classifier, accuracy, hamming-loss and zero-one-loss are equal. Table 4.2 shows the negative p-value criteria for data set C#2. The ABC-accuracy plot in Figure A.2 shows, that the metric distribution follows no probability distribution. Also the MLP-precision plot shows no normal probability distribution.

KS p-value	***acc***	***f1***	***fb***	***hamming***	***jaccard***	***log***	***prec***	***rec***	***zero***	***sum_rows***	***distribution***
rfc	0.236779	0.730176	0.729439	0.13814	0.698578	0.297998	0.845743	0.824085	0.13814	4.639078	weibull_max
knn	0.422616	0.951897	0.980593	0.422574	0.970246	0.690451	0.940962	0.982751	0.422574	6.784663	skewnorm
svm	0.435601	0.478548	0.882289	0.435603	0.601451	0.605145	0.441808	0.886166	0.435603	5.202213	t
dtc	0.492841	0.999108	0.850058	0.492833	0.982112	0.492833	0.977461	0.976476	0.492833	6.756556	beta
gnb	0.591067	0.612275	0.654787	0.557025	0.975491	0.920906	0.440689	0.982824	0.557039	6.292104	exponnorm
lda	0.458651	0.860686	0.393745	0.458651	0.728546	0.759467	0.154682	0.802207	0.458651	5.075287	norm
abc	3.09E-05	0.977633	0.924368	3.09E-05	0.691061	0.449267	0.917066	0.956739	3.09E-05	4.916227	logistic
qda	0.2583	0.953967	0.913645	0.2583	0.852891	0.350437	0.847234	0.715459	0.2583	5.408533	logistic
mlp	0.178113	0.975074	0.925489	0.178113	0.695265	0.832442	3.6E-08	0.674408	0.178113	4.637016	norm
lrc	0.291439	0.827785	0.541725	0.291439	0.880093	0.502971	0.39004	0.662296	0.291439	4.679228	logistic

Table 4.2: The p-values and the chosen distribution of C#2.

Table 4.3 shows the negative p-value criteria for data set C#3. The ABC-accuracy, ABC-hamming-loss, LDA-precision, LRC-fb-score and LRC-precision plot in Figure A.3 shows, that the algorithm-metric distributions follow no probability distribution. ABC-log-loss show similarities with the rayleigh probability distribution. It is possible that a misfit happened.

KS p-value	*acc*	*f1*	*fb*	*hamming*	*jaccard*	*log*	*prec*	*rec*	*zero*	**sum_rows**	**distribution**
rfc	0.194901	0.916991	0.988797	0.208516	0.916821	0.910623	0.45602	0.987497	0.208516	5.788683	burr12
knn	0.123633	0.993243	0.972432	0.113887	0.973963	0.830492	0.818732	0.970476	0.113887	5.910745	exponnorm
svm	0.125382	0.601701	0.920597	0.211812	0.763477	0.722158	0.792089	0.873719	0.211812	5.222746	f
dtc	0.363236	0.878977	0.966899	0.363397	0.535303	0.383744	0.969555	0.803685	0.363397	5.628191	gennorm
gnb	0.270851	0.998373	0.853447	0.299755	0.858112	0.062921	0.073033	0.97712	0.299755	4.693365	loggamma
lda	0.121954	0.992598	0.873546	0.075957	0.999672	0.957551	0.000622	0.886945	0.075957	4.984802	genlogistic
abc	0.02311	0.891657	0.981791	0.005064	0.667961	5.63E-05	0.978897	0.245007	0.005064	3.798608	rayleigh
qda	0.304305	0.743175	0.986806	0.304305	0.934414	0.224697	0.695563	0.750941	0.304305	5.24851	norm
mlp	0.256924	0.699799	0.997811	0.256924	0.920281	0.141807	0.534722	0.472073	0.256924	4.537263	logistic
lrc	0.369639	0.088822	0.026905	0.369639	0.192909	0.49018	0.000993	0.461636	0.369639	2.370361	logistic

Table 4.3: The p-values and the chosen distribution of C#3.

Table 4.4 shows the negative p-value criteria for data set C#4. Figure A.4 shows, that the ABC algorithm have all a logistic probability distribution but at the values higher 0.7 the distribution shows an additional peak. The DTC algorithm show also loggamma probability distribution, but the sparse data density make it hard to fit the distribution in a correct way. MLP algorithm have at all no common probability distribution. The results are random. QDA and RFC algorithm are similar to the DTC algorithm, where the data are to sparse for good fitting. The SVM-precision has no common probability distribution.

KS p-value	*acc*	*f1*	*fb*	*hamming*	*jaccard*	*log*	*prec*	*rec*	*zero*	**sum_rows**	**distribution**
rfc	0.002529	0.648177	0.771426	0.001781	0.891558	0.962075	0.808853	0.991506	0.001781	5.079684	burr12
knn	0.142586	0.931733	0.977299	0.142585	0.936192	0.355545	0.855755	0.723203	0.142588	5.207486	beta
svm	0.135645	0.849677	0.680114	0.135649	0.974484	0.89688	0.031729	0.622203	0.135649	4.462029	beta
dtc	0.004504	0.993715	0.997599	6.62E-06	0.981581	6.62E-06	0.996613	0.958749	6.62E-06	4.932782	loggamma
gnb	0.036248	0.947712	0.945477	0.036247	0.913763	0.589508	0.8839	0.996421	0.036248	5.385524	beta
lda	0.001222	0.762966	0.890237	0.001222	0.825786	0.051845	0.576448	0.474589	0.001222	3.585536	logistic
abc	0.417552	0.000223	0.000269	0.417552	0.01232	0.222116	8.53E-08	0.009035	0.417552	1.496618	logistic
qda	0.000492	0.226978	0.749703	0.000492	0.366216	0.031947	0.808079	0.4829	0.000492	2.667299	logistic
mlp	5.64E-33	NaN	7.66E-30	NaN	3.45E-34	2.98E-06	2.65E-27	1.78E-34	1.12E-07	3.09E-06	wald
lrc	0.052169	0.531656	0.21872	0.052169	0.28168	0.560395	0.156404	0.389363	0.052169	2.294725	logistic

Table 4.4: The p-values and the chosen distribution of C#4.

Table 4.5 shows the negative p-value criteria for data set C#5. The SVM algorithm with negative p-value criteria are not beta distributed. In Figure A.5 one can see that the results of the loop show a probability distribution like beta. Only SVM-recall shows no common distribution. The MLP-algorithm histograms could be described by a logistic probability distribution, so the fitted p-value is real.

KS p-value	*acc*	*f1*	*fb*	*hamming*	*jaccard*	*log*	*prec*	*rec*	*zero*	**sum_rows**	**distribution**
rfc	0.377532	0.954199	0.999984	0.377515	0.965694	0.611735	0.995235	0.853191	0.377515	6.512601	johnsonsb
knn	0.447143	0.997558	0.963923	0.62921	0.98849	0.802221	0.670148	0.981237	0.62921	7.109142	norminvgauss
svm	0.916918	1.42E-10	1.29E-13	0.916916	3.11E-05	0.996092	2.01E-19	1.06E-85	0.916917	3.746874	beta
dtc	0.701162	0.680735	0.89384	0.701162	0.738372	0.701162	0.830248	0.88899	0.701162	6.836835	logistic
gnb	0.831493	0.691683	0.828167	0.831493	0.493957	0.784019	0.730611	0.336022	0.831493	6.358938	logistic
lda	0.662172	0.534186	0.896952	0.814236	0.65121	0.848153	0.099532	0.954187	0.814236	6.274865	exponnorm
abc	0.903965	0.86067	0.83413	0.903965	0.997664	0.950161	0.226359	0.864446	0.903965	7.445325	norm
qda	0.523842	0.970911	0.864815	0.523842	0.950088	0.361544	0.23151	0.620977	0.523842	5.57137	logistic
mlp	0.040994	8.49E-05	0.004062	0.040994	0.000206	0.761444	0.025492	8.51E-12	0.040994	0.914272	logistic
lrc	0.377226	0.605029	0.365018	0.377226	0.932405	0.961125	0.56947	0.419191	0.377226	4.983918	logistic

Table 4.5: The p-values and the chosen distribution of C#5.

Table 4.6 shows the negative p-value criteria for data set C#6. In Figure A.6, the ABC-log-loss shows a logistic probability distribution, but with a low standard deviation. The low p-value equates the real distribution. The KNN-precision is an example why the other precision distributions do not reaching the criteria. The histograms shows no common probability distribution.

KS p-value	*acc*	*f1*	*fb*	*hamming*	*jaccard*	*log*	*prec*	*rec*	*zero*	**sum_rows**	**distribution**
rfc	0.932356	0.965836	0.649814	0.932247	0.945016	0.99689	0.691279	0.851339	0.932247	7.897024	skewnorm
knn	0.669011	0.9133	0.824995	0.66901	0.929595	0.987217	0.002719	0.929639	0.66901	6.594497	gennorm
svm	0.562352	0.732459	0.815636	0.6509	0.855912	0.958847	0.001618	0.604047	0.6509	5.832671	gamma
dtc	0.639982	0.799687	0.983697	0.643317	0.915409	0.626159	0.905314	0.7604	0.64514	6.919105	skewnorm
gnb	0.916433	0.799318	0.980996	0.916432	0.723432	0.811044	0.871193	0.600609	0.916432	7.53589	beta
lda	0.887719	0.912305	0.930502	0.887719	0.981743	0.331163	0.99663	0.903868	0.887719	7.719366	norm
abc	0.911338	0.677632	0.965003	0.911338	0.977608	0.005919	0.796877	0.407187	0.911338	6.564241	logistic
qda	0.349784	0.963602	0.818277	0.349784	0.796336	0.688117	0.048	0.195086	0.349784	4.55877	logistic
mlp	0.570319	0.347745	0.170447	0.570319	0.285596	0.689153	0.615077	0.335033	0.570319	4.154009	logistic
lrc	0.616561	0.48336	0.826029	0.616561	0.55116	0.373035	0.909656	0.672655	0.616561	5.665578	norm

Table 4.6: The p-values and the chosen distribution of C#6.

Table 4.7 shows the negative p-value criteria for data set C#7. In Figure A.7 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution.

KS p-value	*acc*	*f1*	*fb*	*hamming*	*jaccard*	*log*	*prec*	*rec*	*zero*	**sum_rows**	**distribution**
rfc	1.17E-83	3.07E-86	1.17E-89	9.7E-107	1.15E-78	0.90029	1.43E-86	1.47E-80	1.3E-117	0.90029	invgamma
knn	6.89E-38	1.18E-42	7.05E-63	1.5E-101	NaN	0.994528	5.3E-43	NaN	NaN	0.994528	johnsonsb
svm	0.60386	0.615581	0.357817	0.603821	0.564494	0.537646	0.396105	0.973999	0.603821	5.257144	skewnorm
dtc	1.76E-53	3.25E-89	2.66E-52	NaN	5.98E-82	6.75E-07	4.9E-49	3.28E-49	3.71E-86	6.75E-07	t
gnb	0.250965	0.64485	0.805845	0.201457	0.852404	0.927562	0.981968	0.627977	0.201457	5.494484	gamma
lda	0.330626	0.751709	0.597549	0.330623	0.921534	0.975595	0.724707	0.867683	0.330623	5.830647	beta
abc	0.315205	0.968927	0.951374	0.298521	0.926438	0.675836	0.926438	NaN	0.298521	5.361261	genlogistic
qda	0.172221	0.830813	0.714404	0.172221	0.41384	0.960201	0.421158	0.3019	0.172221	4.158979	logistic
mlp	0.152663	0.183429	0.089014	0.152663	0.527191	0.501892	0.007632	0.019754	0.152663	1.786902	logistic
lrc	0.18407	0.689785	0.703798	0.18407	0.795343	0.03502	0.706292	0.575526	0.18407	4.057974	logistic

Table 4.7: The p-values and the chosen distribution of C#7.

Table 4.8 shows the negative p-value criteria for data set R#1. In Figure A.8 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution. Only the MLP-r2 and the DTR-r2 could be described by inverse gamma and skewnorm probability distribution. So the low p-value fitted by the package is trust-able.

KS p-value	**mer**	**mae**	**mse**	**msle**	**mee**	**r2**	**sum_rows**	**distribution**
lir	4.76E-05	0.971675	0.880751	0.98036	0.968456	0.925566	4.726856	genlogistic
las	0.000143	0.858197	0.919099	0.918009	0.988635	0.761496	4.445578	burr12
svr	2.65E-11	0.996677	0.537335	0.870867	0.915548	0.791199	4.111627	skewnorm
rid	5E-05	0.971295	0.880437	0.980057	0.967278	0.925277	4.724394	genlogistic
enc	9.56E-05	0.763221	0.753573	0.966068	0.8915	0.962133	4.33659	genlogistic
mlp	0.553692	0.670775	0.533055	0.994147	0.605352	NaN	3.357021	invgamma
dtr	0.000256	0.727302	0.938244	0.947733	5.64E-15	3.65E-05	2.61357	skewnorm
rfr	6.25E-06	0.838852	NaN	0.845236	0.998861	0.376532	3.059487	skewnorm
knn	NaN	0.935514	0.290581	0.636907	0.128683	0.970824	2.96251	genlogistic

Table 4.8: The p-values and the chosen distribution of R#1.

Table 4.9 shows the negative p-value criteria for data set R#2. In Figure A.9 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution. Only KNN-mer shows approximating a generalized normal distribution. The p-value is trust-able. DTR-mee is not fit-able because of sparse data set.

KS p-value	mer	mae	mse	msle	mee	r2	sum_rows	distribution
lir	9.76E-08	0.984139	0.939401	0.7998	0.325616	0.987885	4.036841	beta
las	8.92E-08	0.985705	0.94566	0.801113	0.314321	0.988358	4.035157	beta
svr	0	0.822944	0.826051	0.591481	0.771788	0.846788	3.859051	genlogistic
rid	9.76E-08	0.984249	0.939413	0.7998	0.322955	0.987884	4.034301	beta
enc	7.5E-06	0.963628	0.825676	0.713737	0.820744	0.843571	4.167364	burr12
mlp	0.001769	0.887085	0.995981	0.873132	0.947631	0.694528	4.400127	beta
dtr	0.086733	0.602758	0.46451	0.610289	0.002474	0.407751	2.174514	genlogistic
rfr	0.071777	0.993488	0.969736	0.92921	0.91717	0.860164	4.741546	johnsonsb
knn	0.001286	0.918747	0.883666	0.408028	0.302974	0.965482	3.480183	gennorm

Table 4.9: The p-values and the chosen distribution of R#2.

Table 4.10 shows the negative p-value criteria for data set R#3. In Figure A.10 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution. Only the R2 algorithm metrics could be described by right- and left-aligned probability distributions. The p-values for the negative p-value criteria histograms of R2 are trust-able.

KS p-value	mer	mae	mse	msle	mee	r2	sum_rows	distribution
lir	0.394158	0.956029	0.821213	0.993026	0.726032	7.11E-07	3.890457	burr12
las	0.057147	0.983387	0.853169	0.97837	0.947818	0.033166	3.853057	johnsonsu
svr	4.83E-18	0.753468	0.034745	0.83861	0.656818	0.496831	2.780472	nakagami
rid	0.279148	0.925846	0.775652	0.975561	0.830065	4.37E-09	3.786273	burr12
enc	0.003101	0.690724	0.859179	0.83876	0.984263	9.73E-06	3.376036	mielke
mlp	0.059275	0.577093	0.399032	0.927236	0.345946	0.085416	2.393998	mielke
dtr	0.000155	0.957742	0.730232	0.440507	0.337108	NaN	2.465744	mielke
rfr	3.55E-05	0.653488	0.024797	0.276842	0.907684	0.006935	1.869782	burr
knn	NaN	0.871713	0.429561	0.993647	0.536536	0.727869	3.559326	mielke

Table 4.10: The p-values and the chosen distribution of R#3.

Table 4.11 shows the negative p-value criteria for data set R#4. In Figure A.11 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution. All histograms show random distributed values with no probability.

KS p-value	mer	mae	mse	msle	mee	r2	sum_rows	distribution
lir	1.11E-07	0.826488	0.806131	0.986137	0.958952	0.970107	4.547815	skewnorm
las	3.74E-07	0.813352	0.832004	0.8829	0.718076	0.971776	4.218109	genlogistic
svr	2.06E-18	0.899939	0.906075	0.896499	0.945556	0.740812	4.388881	beta
rid	7.63E-08	0.926105	0.797583	0.851331	0.914994	0.980306	4.470319	skewnorm
enc	5.6E-08	0.754745	0.818384	0.968244	0.832933	0.781403	4.15571	genlogistic
mlp	4.45E-07	0.932181	0.917257	0.909	0.868165	0.787663	4.414267	skewnorm
dtr	5.41E-09	0.947645	0.997151	0.86262	0.263178	0.965576	4.036169	genlogistic
rfr	3.67E-06	0.918334	0.662312	0.834865	0.864208	0.98788	4.267602	skewnorm
knn	7.41E-06	0.487407	0.537748	0.715585	0.850577	0.781657	3.372981	exponnorm

Table 4.11: The p-values and the chosen distribution of R#4.

Table 4.12 shows the negative p-value criteria for data set R#5. In Figure A.12 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution. The LIR-msle, RID-msle and ENC-msle histograms show left-aligned distributions, so the p-value is trust-able.

KS p-value	mer	mae	mse	msle	mee	r2	sum_rows	distribution
lir	3.54E-08	0.965254	0.933981	2.74E-10	0.889558	0.900499	3.689292	johnsonsu
las	0.007924	0.924878	0.856319	0.406695	0.300095	0.992931	3.488842	johnsonsu
svr	0.003516	0.731014	0.864272	0.91282	0.476257	0.977307	3.965186	genlogistic
rid	0.000132	0.784526	0.445139	0.000113	0.990646	0.899074	3.11963	weibull_min
enc	0.78853	0.481528	0.870103	0.00512	0.427768	0.972062	3.545111	genlogistic
mlp	0.012646	0.834455	0.823287	0.997767	0.958667	0.897669	4.52449	burr12
dtr	0.386687	0.585021	0.853321	0.950138	0.360246	0.998126	4.133539	genlogistic
rfr	0.943795	0.795287	0.240578	NaN	0.934502	0.949572	3.863733	genlogistic
knn	0.021385	0.821087	0.942787	0.971591	0.663269	0.900084	4.320203	genlogistic

Table 4.12: The p-values and the chosen distribution of R#5.

Table 4.13 shows the negative p-value criteria for data set R#6. In Figure A.13 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution. The KNN-mee histogram shows a skewnorm distribution, but the data are too sparse.

KS p-value	mer	mae	mse	msle	mee	r2	sum_rows	distribution
lir	4.36E-09	0.876945	0.759472	0.725052	0.985555	0.913907	4.260931	burr12
las	0	0.817949	0.960104	0.705183	0.827122	0.88952	4.199878	powernorm
svr	1.09E-30	0.91857	0.663466	0.945965	0.763329	0.714485	4.005815	beta
rid	6.78E-77	0.808636	0.597068	0.965368	0.832605	0.94769	4.151366	beta
enc	4.51E-89	0.781198	0.697022	0.963537	0.999979	0.989779	4.431515	beta
mlp	NaN	0.997201	0.995103	0.987218	0.834535	0.845176	4.659233	beta
dtr	NaN	0.993853	0.610842	0.854023	1.49E-12	0.790652	3.249371	beta
rfr	0.00017	0.624637	0.66436	0.976823	0.561325	0.932531	3.759846	skewnorm
knn	0.000314	0.894577	0.846353	0.973941	0.004579	0.99998	3.719744	skewnorm

Table 4.13: The p-values and the chosen distribution of R#6.

Table 4.14 shows the negative p-value criteria for data set R#7. In Figure A.14 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution.

KS p-value	mer	mae	mse	msle	mee	r2	sum_rows	distribution
lir	0.036072	0.876563	0.512755	0.815288	0.957729	0.935203	4.133611	loggamma
las	0.001124	0.944134	0.771946	0.818814	0.999654	0.967558	4.50323	beta
svr	5.98E-05	0.963966	0.882417	0.990818	0.848692	0.989293	4.675245	beta
rid	0.001107	0.849709	0.638703	0.997704	0.591852	0.96357	4.042645	burr12
enc	0.000955	0.897137	0.752135	0.917028	0.9898	0.867698	4.424753	beta
mlp	0.000116	0.934655	0.561697	0.592995	0.856243	0.957965	3.903671	burr12
dtr	1.41E-07	0.94517	0.928546	0.971239	0.723243	0.997192	4.565391	burr12
rfr	0.49824	0.959156	0.9473	0.901441	0.805303	0.556728	4.668168	burr12
knn	0.032457	0.558924	0.586566	0.726943	0.803951	0.461127	3.169968	skewnorm

Table 4.14: The p-values and the chosen distribution of R#7.

Table 4.15 shows the negative p-value criteria for data set R#8. The histograms are not plotted in the annex, they are stored on the Github server stated before. All algorithm-metric results with a negative p-value criteria show no common probability distribution.

KS p-value	mer	mae	mse	msle	mee	r2	sum_rows	distribution
lir	0.000129	6.11E-06	0.004506	0.002667	6.11E-06	1.99E-63	0.007314	invweibull
las	2.13E-29	1.94E-05	6.77E-28	5.08E-23	1.94E-05	3.06E-23	3.89E-05	beta
svr	2.05E-07	7.26E-10	1.32E-38	1.11E-06	7.26E-10	1.47E-43	1.32E-06	skewcauchy
rid	3.91E-06	NaN	0.000346	0.000773	NaN	NaN	0.001123	geninvgauss
enc	8.3E-11	1.94E-06	2.02E-15	3.52E-13	1.94E-06	3.59E-43	3.88E-06	levy
mlp	1.41E-06	0.026076	0.022742	9.17E-05	0.026076	9.96E-10	0.074986	skewcauchy
dtr	9.71E-24	2.16E-26	5.84E-27	1.31E-29	2.16E-26	NaN	9.76E-24	genlogistic
rfr	NaN	NaN	NaN	NaN	NaN	NaN	0	johnsonsb
knn	8.76E-06	5.89E-07	1.9E-08	1.02E-07	NaN	7.74E-08	9.55E-06	skewcauchy

Table 4.15: The p-values and the chosen distribution of R#8.

Table 4.16 shows the negative p-value criteria for data set R#9. In Figure A.15 one can see, that all algorithm-metric results with a negative p-value criteria show no common probability distribution.

KS p-value	mer	mae	mse	msle	mee	r2	sum_rows	distribution
lir	4.66E-08	0.780292	0.324148	0.018329	0.736459	0.004303	1.863531	skewnorm
las	2.64E-33	0.771123	7.18E-05	0.003994	0.876193	1.61E-05	1.651398	beta
svr	0.008118	0.898792	0.296604	0.94329	0.959033	0.163533	3.26937	beta
rid	2.07E-08	0.872863	0.160968	0.00315	0.987992	3E-08	2.024974	nakagami
enc	4.72E-14	0.860915	0.002059	3.58E-07	0.873281	9.45E-14	1.736256	beta
mlp	0.003985	0.982034	0.946104	0.976065	0.83765	0.792432	4.538269	skewnorm
dtr	7.72E-12	0.997244	0.241646	0.467813	0.155191	2.41E-06	1.861896	exponnorm
rfr	2.93E-07	0.878832	0.014669	6.71E-05	0.916533	6.21E-05	1.810163	exponnorm
knn	1.1E-05	0.845874	0.320779	0.832242	0.633973	0.694695	3.327572	gamma

Table 4.16: The p-values and the chosen distribution of R#9.

4.2 Normalized standard deviation

With the output parameters of the fitted distributions in the section before, the normalized standard deviation is calculated. The negative criteria for the normalized standard deviation $s_n > 0.5$ enable only a small spreading of the distributions. The cells in red show the negative criteria of the data set. The green cell border implements the algorithm-metric s_n with a negative p-value criteria.

Table 4.17 is the normalized standard deviation of the data C#1. The ABC-log, with a NaN p-value, has also no standard deviation. The p-value = 0.000557 of the ABC-prec and the p-value = 0.01334 of the ABC-hamming/ABC-zero implies that s_n is not trust-able for the interpretation.

standard deviation	acc	f1	fb	hamming	jaccard	log	prec	rec	zero
rfc	0.055976	0.10953	0.116305	0.072016	0.123426	0.204599	0.140837	0.105894	0.072016
knn	0.064754	0.104066	0.112711	0.072042	0.118419	0.162536	0.141139	0.099908	0.072042
svm	0.074246	0.124953	0.13978	0.075	0.133199	0.042221	0.16414	0.108536	0.075
dtc	0.080118	0.12204	0.127007	0.074588	0.140176	0.074588	0.138894	0.126498	0.074588
gnb	0.074783	0.109771	0.123191	0.057493	0.13266	0.150845	0.134011	0.090124	0.057493
lda	0.06228	0.103226	0.109536	0.074889	0.11643	0.091086	0.130282	0.100176	0.074889
abc	0.117822	0.179298	0.287938	*0.314322*	0.164237	*NaN*	*0.735793*	0.124503	*0.314322*
qda	0.065	0.09691	0.100295	0.06837	0.110653	0.195408	0.107321	0.105011	0.06837
mlp	0.06247	0.123747	0.133552	0.070008	0.132448	0.078835	0.162274	0.115811	0.070008
lrc	0.069131	0.121666	0.130496	0.077272	0.133416	0.051804	0.158396	0.109219	0.077272

Table 4.17: The normalized standard deviation of C#1.

Table 4.18 is the normalized standard deviation of the data C#2. The s_n is for all algorithm-metric distribution < 0.5 and only four of them are not reaching the p-value criteria. The s_n for hamming and zero are for multi class predictor equal. The results are highly use-able for the interpretation of the stability of the metrics, because most of the p-values are > 0.05.

standard deviation	**acc**	**f1**	**fb**	**hamming**	**jaccard**	**log**	**prec**	**rec**	**zero**
rfc	0.037372	0.061087	0.061865	0.101192	0.077494	0.220847	0.067984	0.064237	0.101192
knn	0.05208	0.08086	0.084388	0.069246	0.093537	0.16917	0.09519	0.07977	0.069246
svm	0.056945	0.096607	0.11343	0.079815	0.110239	0.040385	0.161883	0.078224	0.079815
dtc	0.047081	0.063417	0.063693	0.094231	0.082644	0.094231	0.065117	0.06753	0.094231
gnb	0.05533	0.066101	0.073665	0.070761	0.081158	0.094991	0.103471	0.061735	0.070761
lda	0.045803	0.068315	0.081285	0.089981	0.08331	0.072523	0.065786	0.049152	0.089981
abc	*0.188307*	0.12342	0.125284	*0.210155*	0.1569	0.035576	0.134485	0.111331	*0.210155*
qda	0.051578	0.066948	0.067065	0.077089	0.082949	0.1176	0.068063	0.064318	0.077089
mlp	0.088235	0.109948	0.152169	0.118051	0.126348	0.063452	*0.349067*	0.082804	0.118051
lrc	0.056276	0.064876	0.080905	0.074829	0.086602	0.039288	0.112713	0.055016	0.074829

Table 4.18: The normalized standard deviation of C#2.

Table 4.19 is the normalized standard deviation of the data C#3. Like for data set C#2, also these results are highly use-able for the selection of the stability of the metrics. Here the criteria of the p-value will be the basic for the recommendation of the reliability of the data.

standard deviation	**acc**	**f1**	**fb**	**hamming**	**jaccard**	**log**	**prec**	**rec**	**zero**
rfc	0.080839	0.100849	0.10362	0.065771	0.111902	0.178516	0.113225	0.099041	0.065771
knn	0.087272	0.109503	0.118345	0.073107	0.127539	0.117241	0.138577	0.103337	0.073107
svm	0.085332	0.107167	0.134436	0.060934	0.123381	0.036048	0.19736	0.083679	0.060934
dtc	0.090684	0.109857	0.112777	0.063002	0.123935	0.062954	0.118657	0.110038	0.063002
gnb	0.093449	0.130789	0.154639	0.064259	0.158246	0.091249	0.203245	0.100326	0.064259
lda	0.103159	0.178051	0.240048	0.06063	0.197042	0.048342	*0.333728*	0.109462	0.06063
abc	*0.363928*	0.333977	0.350282	*0.09202*	0.36453	*0.08663*	0.377786	0.329436	*0.09202*
qda	0.089825	0.122286	0.148601	0.062156	0.141869	0.096097	0.206538	0.09404	0.062156
mlp	0.096237	0.115598	0.144683	0.06203	0.127894	0.089875	0.213207	0.095207	0.06203
lrc	0.135557	0.328622	*0.383782*	0.066846	0.369399	0.039886	*0.429883*	0.190425	0.066846

Table 4.19: The normalized standard deviation of C#3.

Table 4.20 is the normalized standard deviation of the data C#4. The s_n over-all MLP-metrics and the QDA-log reach the negative criteria. For the MLP the negative p-value criteria describes the s_n value. The QDA-log distribution is strongly scattered. 1/3 of the p-values do not reach the >0.05 criteria, so the data set is less valuable for the recommendation of metric stability.

standard deviation	**acc**	**f1**	**fb**	**hamming**	**jaccard**	**log**	**prec**	**rec**	**zero**
rfc	*0.036651*	0.032115	0.032209	*0.417349*	0.054205	0.229512	0.031785	0.030369	*0.417349*
knn	0.096997	0.102952	0.088314	0.138879	0.14823	0.414182	0.071958	0.106092	0.138879
svm	0.15317	0.227565	0.223949	0.117307	0.256029	0.050777	*0.19998*	0.149981	0.117307
dtc	*0.041884*	0.036811	0.035813	*0.381243*	0.060272	*0.381243*	0.035009	0.037388	*0.381243*
gnb	*0.042364*	0.037761	0.03793	*0.256433*	0.060245	0.282413	0.037583	0.036091	*0.256433*
lda	*0.051842*	0.042922	0.042385	*0.209865*	0.059589	0.273401	0.043013	0.044988	*0.209865*
abc	0.106415	*0.111152*	*0.134729*	0.21467	*0.117123*	0.175279	*0.183421*	*0.091928*	0.21467
qda	*0.03726*	0.03237	0.032745	0.38121	0.054808	0.514061	0.032016	0.029008	*0.38121*
mlp	*0.806459*	*0.901193*	*0.930722*	*0.965673*	*0.991827*	*0.857828*	*0.951307*	*0.762152*	*0.965673*
lrc	0.11267	0.129708	0.13157	0.17369	0.146542	0.077147	0.12596	0.109143	0.17369

Table 4.20: The normalized standard deviation of C#4.

Table 4.21 is the normalized standard deviation of the data C#5. The s_n ABC metrics tend to higher values. 13 out of 90 p-values reach the negative criteria. So the data set is good for the recommendation of the algorithm-metric context based on the normalized standard deviation.

standard deviation	**acc**	**f1**	**fb**	**hamming**	**jaccard**	**log**	**prec**	**rec**	**zero**
rfc	0.039155	0.069289	0.074956	0.032515	0.073722	0.110448	0.087536	0.066629	0.032515
knn	0.040432	0.071447	0.096113	0.027398	0.072097	0.066442	0.154163	0.055547	0.027398
svm	0.037385	*0.054081*	*0.080356*	0.027761	*0.059807*	0.019613	*0.10554*	*0.010806*	0.027761
dtc	0.04816	0.076575	0.078858	0.032111	0.083383	0.032111	0.083213	0.080644	0.032111
gnb	0.042479	0.102002	0.10559	0.033341	0.106115	0.02341	0.128028	0.064956	0.033341
lda	0.039644	0.057403	0.07254	0.030289	0.065161	0.019049	0.145449	0.042601	0.030289
abc	0.149714	0.112499	0.11931	0.072981	0.117764	0.006801	0.159092	0.106583	0.072981
qda	0.039119	0.065274	0.086779	0.032643	0.067825	0.027893	0.129705	0.048384	0.032643
mlp	*0.072721*	*0.263725*	*0.333267*	*0.051931*	*0.223118*	0.027682	*0.435756*	*0.13146*	*0.051931*
lrc	0.040005	0.087923	0.147124	0.028926	0.076862	0.017	0.229965	0.025449	0.028926

Table 4.21: The normalized standard deviation of C#5.

Table 4.22 is the normalized standard deviation of the data C#6. ABC-log, RFC-, KNN- and QDA-prec are based on the negative p-value criteria. At all the data set is highly use-able for the recommendation of the algorithm-metric relation.

standard deviation	**acc**	**f1**	**fb**	**hamming**	**jaccard**	**log**	**prec**	**rec**	**zero**
rfc	0.021165	0.057287	0.065791	0.032797	0.058493	0.083851	*0.095477*	0.052003	0.032797
knn	0.024718	0.058527	0.078721	0.022746	0.056611	0.051349	*0.147339*	0.048991	0.022746
svm	0.023702	0.076988	0.108432	0.023664	0.070955	0.018763	0.31113	0.041884	0.023664
dtc	0.028666	0.069416	0.069631	0.029806	0.07008	0.029818	0.073675	0.077511	0.029797
gnb	0.057941	0.083102	0.100038	0.031419	0.084085	0.044569	0.160604	0.094522	0.031419
lda	0.023582	0.048631	0.052623	0.024271	0.055089	0.031046	0.060183	0.046806	0.024271
abc	0.252613	0.243364	0.22885	0.086344	0.255923	*0.034944*	0.202352	0.207907	0.086344
qda	0.045088	0.08158	0.090782	0.026257	0.083374	0.049428	*0.149743*	0.080341	0.026257
mlp	0.029832	0.217903	0.178538	0.030906	0.220549	0.040102	0.174125	0.213969	0.030906
lrc	0.023166	0.053614	0.06028	0.024705	0.054531	0.017292	0.080019	0.046307	0.033255

Table 4.22: The normalized standard deviation of C#6.

Table 4.23 is the normalized standard deviation of the data C#7. Around 1/3 of the p-values reach negative criteria. All negative s_n criteria are based on negative p-value criteria.The results show that the recommendation based on this data set is driven by p-value and s_n.

standard deviation	**acc**	**f1**	**fb**	**hamming**	**jaccard**	**log**	**prec**	**rec**	**zero**
rfc	*0.00233*	*0.001959*	*0.001601*	*-inf*	*0.00408*	-0.37274	*0.001415*	*0.002488*	*-inf*
knn	*0.037847*	*0.05977*	*0.025488*	*1.406257*	*0.064553*	0.525843	*0.04242*	*0.055448*	*1.406253*
svm	0.033379	0.033038	0.035133	0.233927	0.058878	0.140794	0.035754	0.026769	0.233927
dtc	*3.13E-06*	*NaN*	*0.005866*	*6.548546*	*0.007184*	*0.004462*	*NaN*	*NaN*	*6.548546*
gnb	0.053121	0.051256	0.05248	0.174142	0.078663	0.250068	0.049966	0.039539	0.174142
lda	0.03968	0.038069	0.037086	0.115281	0.061026	0.064251	0.037447	0.041968	0.115281
abc	0.034751	0.018582	0.02704	0.086288	0.031973	0.00037	0.031973	*3.34E-22*	0.086288
qda	0.049047	0.061311	0.069611	0.137523	0.08666	0.122038	0.044091	0.045492	0.137523
mlp	0.027605	0.054845	0.047704	0.438445	0.088872	0.233852	*0.042826*	*0.062314*	0.438445
lrc	0.023828	0.034635	0.033786	0.20788	0.053621	*0.093584*	0.034294	0.037548	0.20788

Table 4.23: The normalized standard deviation of C#7.

Table 4.24 is the normalized standard deviation of the data R#1. Beside the mer, the results are based on positive p-value criteria. The only negative s_n criteria is not trust-able because of negative p-value criteria. The data set is dominated by the s_n and so the recommendation is mainly influeced by this value.

standard deviation	mer	mae	mse	msle	mee	r2
lir	*0.29052*	0.101738	0.278022	0.190694	0.147567	0.153469
las	*0.273824*	0.10394	0.28964	0.185631	0.141102	0.152112
svr	*0.343976*	0.123824	0.361241	0.183287	0.138341	0.135582
rid	*0.290826*	0.101739	0.27803	0.190691	0.147563	0.153468
enc	*0.268392*	0.103862	0.269121	0.187971	0.158204	0.144488
mlp	0.407687	0.124194	0.380329	0.368882	0.145088	*0.275252*
dtr	*0.394177*	0.125707	0.370739	0.189699	*0.15982*	*0.866403*
rfr	*0.387491*	0.114617	*0.345816*	0.205218	0.128553	0.161104
knn	*0.36286*	0.106496	0.335057	0.192267	0.136249	0.111909

Table 4.24: The normalized standard deviation of R#1.

Table 4.25 is the normalized standard deviation of the data R#2. Beside the mer, the results are based on positive p-value criteria. The negative s_n of DTR-r2 is based on a negative mean of the r2 metric. The data set is highly use-able for recommendation based on the s_n

standard deviation	mer	mae	mse	msle	mee	r2
lir	*0.06681*	0.031379	0.105046	0.15281	0.039834	0.135019
las	*0.066707*	0.031372	0.105013	0.152802	0.03968	0.134705
svr	*0.039065*	0.033757	0.098095	0.149579	0.031857	0.114041
rid	*0.06681*	0.031381	0.105046	0.15281	0.039832	0.135018
enc	*0.056775*	0.032004	0.107294	0.152215	0.040809	0.127766
mlp	*0.043565*	0.032737	0.11122	0.161076	0.040789	0.096724
dtr	0.054812	0.039497	0.101692	0.145068	*0.047179*	-1.10726
rfr	0.068765	0.025467	0.106708	0.167146	0.037726	0.10432
knn	*0.077931*	0.027089	0.093472	0.155913	0.034369	0.128748

Table 4.25: The normalized standard deviation of R#2.

Table 4.26 is the normalized standard deviation of the data R#3. Only a few p-value reach the negative criteria. MSE shows over-all $s_n > 0.5$. Most of the negative s_n criteria are based on positive p-value criteria. So the data set is use-able for the recommendation based on the s_n.

standard deviation	mer	mae	mse	msle	mee	r2
lir	0.379323	0.193289	0.503186	0.468667	0.264604	*0.303063*
las	0.489541	0.189946	0.585456	0.453853	0.211148	*0.601712*
svr	*0.439699*	0.279329	*0.662278*	0.320819	NaN	NaN
rid	0.486793	0.192416	0.56822	0.495794	0.248886	*0.288088*
enc	*0.913264*	0.209755	0.900346	0.566349	0.200841	*NaN*
mlp	0.364799	0.244276	0.547318	0.49509	0.504332	NaN
dtr	*0.409411*	0.313967	0.543758	0.411549	0.437757	*-6.5E-06*
rfr	*0.509899*	0.39456	*2.255789*	0.430837	0.458518	*0.186703*
knn	*NaN*	0.407128	inf	0.471667	0.367832	NaN

Table 4.26: The normalized standard deviation of R#3.

Table 4.27 is the normalized standard deviation of the data R#4. The three negative s_n criteria are based on positive p-value criteria. The data set is highly use-able for the recommendation based on the s_n criteria.

standard deviation	**mer**	**mae**	**mse**	**msle**	**mee**	**r2**
lir	*0.284155*	0.051738	0.158375	0.116341	0.067704	0.504653
las	*0.308531*	0.052658	0.158441	0.114607	0.069782	0.381729
svr	*0.233201*	0.068421	0.175989	0.088732	0.056944	4.949006
rid	*0.283405*	0.051688	0.157803	0.113012	0.068566	0.489343
enc	*0.297339*	0.052196	0.157796	0.112242	0.06554	0.342169
mlp	*0.342421*	0.055582	0.187452	0.116372	0.075386	0.272173
dtr	*0.204458*	0.094672	0.205237	0.143752	0.137693	-2.21527
rfr	*0.363943*	0.069231	0.233348	0.113409	0.090208	0.231218
knn	*0.301405*	0.065437	0.217859	0.104638	0.098044	0.27225

Table 4.27: The normalized standard deviation of R#4.

Table 4.28 is the normalized standard deviation of the data R#5. Two out of three negative s_n of the msle are related to the negative p-value criteria. Again the data set has a high quality for recommendation based on the s_n.

standard deviation	**mer**	**mae**	**mse**	**msle**	**mee**	**r2**
lir	*0.299759*	0.115752	0.265862	*6.657807*	0.182362	0.022769
las	*0.134913*	0.108226	0.201083	0.56616	0.177986	0.030131
svr	*0.189635*	0.116513	0.235764	0.234297	0.133522	0.248865
rid	*0.188035*	0.104398	0.197419	*0.604153*	0.166201	0.033696
enc	0.122522	0.109793	0.204411	*0.391854*	0.163686	0.035568
mlp	*0.200685*	0.117814	0.230451	0.404617	0.168879	0.024971
dtr	0.212072	0.154151	0.297668	0.314812	0.213266	0.105603
rfr	0.195863	0.155856	0.321283	0.372076	0.196292	0.033489
knn	*0.181704*	0.116404	0.21911	0.260575	0.170074	0.176561

Table 4.28: The normalized standard deviation of R#5.

Table 4.29 is the normalized standard deviation of the data R#6. The data set shows similar results to the R#5 with similar quality for the recommendation. The two negative s_n criteria are based on a negative p-value criteria.

standard deviation	mer	mae	mse	msle	mee	r2
lir	0.243829	0.039106	0.078317	0.099524	0.059475	0.009361
las	NaN	0.040968	0.078839	0.105813	0.067847	0.009602
svr	0.507276	0.050212	0.157697	0.148317	0.069007	0.014013
rid	0.345519	0.039034	0.076918	0.097805	0.059698	0.00916
enc	0.328004	0.038635	0.075043	0.095186	0.064133	0.009284
mlp	0.25797	0.040748	0.081946	0.096825	0.059505	0.010251
dtr	0.317557	0.051994	0.106357	0.106254	0.088166	0.017821
rfr	0.3177	0.048663	0.119607	0.110355	0.073971	0.009589
knn	0.317719	0.051276	0.125743	0.13146	0.075795	0.012203

Table 4.29: The normalized standard deviation of R#6.

Table 4.30 is the normalized standard deviation of the data R#7. The data set has no negative s_n criteria and only the mer metric shows negative p-value criteria.

standard deviation	mer	mae	mse	msle	mee	r2
lir	0.127923	0.071409	0.167865	0.145593	0.067032	0.05841
las	0.126978	0.075076	0.163111	0.213576	0.101042	0.064513
svr	0.195733	0.089718	0.24207	0.12051	0.065761	0.12374
rid	0.109444	0.073754	0.169767	0.146172	0.068878	0.059543
enc	0.123906	0.075094	0.163163	0.215719	0.102915	0.065574
mlp	0.116186	0.076593	0.165684	0.278061	0.124747	0.061869
dtr	0.172367	0.095917	0.218865	0.187778	0.142596	0.145278
rfr	0.196712	0.088452	0.209498	0.169524	0.110678	0.070551
knn	0.185083	0.077459	0.172351	0.150669	0.105946	0.091748

Table 4.30: The normalized standard deviation of R#7.

Table 4.31 is the normalized standard deviation of the data R. All p-values and most of the s_n values show negative criteria. For this reason the data set will not be used for the recommendation of the algorithm-metrics relation.

standard deviation	mer	mae	mse	msle	mee	r2
lir	0.213799	0.240207	0.119523	0.032134	0.240207	NaN
las	0.562711	0.775496	0.966436	1.227257	0.775496	-2.42839
svr	45.40154	0.575481	4.871285	2.118756	0.575481	NaN
rid	NaN	1.84E-12	NaN	NaN	1.84E-12	-1.4E-13
enc	-0.63922	NaN	NaN	-4.02772	NaN	NaN
mlp	NaN	NaN	8.19E-05	0.33252	NaN	NaN
dtr	0.671902	0.691067	0.766903	0.828223	0.691067	-1.1698
rfr	0.646356	0.628681	0.982015	1.033859	0.628681	-6.8982
knn	3.55E+90	295.9816	330.5595	1081.443	295.9816	NaN

Table 4.31: The normalized standard deviation of R#8.

Table 4.32 is the normalized standard deviation of the data R. Around 40% of the results reach negative p-value criteria. All mer are negative. Half of the negative s_n criteria are based on negative p-values. So the data set has limited useability for recommendation.

standard deviation	**mer**	**mae**	**mse**	**msle**	**mee**	**r2**
lir	*0.470544*	0.157709	0.541559	*0.442717*	0.175139	*0.012582*
las	*0.713861*	0.175848	*0.525232*	*0.394249*	0.176843	*0.012666*
svr	*0.464136*	0.213729	0.654192	0.453029	0.176498	0.028767
rid	*0.46918*	0.154781	0.542401	*0.439443*	0.166927	*NaN*
enc	*0.62822*	0.172495	*0.553256*	*0.471707*	0.162386	*0.0149*
mlp	*0.29196*	0.179433	0.415003	0.15156	0.190417	-0.35
dtr	*0.366827*	0.205429	0.618641	0.408734	0.171335	*0.039553*
rfr	*0.497487*	0.192672	*0.767002*	*0.588627*	0.177871	*0.021984*
knn	*0.336442*	0.138613	0.380925	0.267929	0.191598	0.02702

Table 4.32: The normalized standard deviation of R#9.

5 Interpretation and Recommendation

The results discussed in chapter 4 are the basics for the recommendation, which algorithm-metric relation is useful in geotechnic. First an over-all recommendation is given, where the sum of all negative p-value criteria and negative s_n criteria gives the hint for the decision. In a second step every type of data set is discussed separately.

Table 5.1 is the recommendation for regessors. The criteria for green (+; use-able for all data sets), orange ($\sim$; depends on data set), red (- not use-able for all data sets) is based on the p-value and s_n. For regression, results of nine data sets are available. First the negative p-value criteria is summed up. The negative s_n criteria counts double if it is not based on a negative p-value criteria, else it will not be summed into the recommendation. The sum is refered to the number of data sets. Equation 5.1 shows the summation criteria for the regressor. If the value reaches >30% the over-all recommendation is $\sim$. For values > 50% it is $-$.

$$Regressor_{crit} = \frac{\sum pvalue_{neg} + 2 * s_n(pvalue_{pos})}{n_{datasets}} * 100 \tag{5.1}$$

The results show for mer a negative over-all recommendation. This is based on the negative p-value criteria, for mer between 7-9 times of the cycle. mse is mostly acceptable, but for LIR, RID and DTR not use-able. Here the 1-2 negative s_n criteria have an impact. The same is for r2, but here the p-value has more impact. MEE and mae are over-all metrics use-able. The SVR-mee and MLP-mee shows a $\sim$ criteria because of one time negative p-value and s_n criteria. LIR-msle and RID-msle is based on three p-value criteria, LAS-msle one s_n and two p-value, ENC-msle one s_n and three p-value criteria.

recommendation	mer	mae	mse	msle	mee	r2
lir	-	+	-	~	+	-
las	-	+	~	~	+	~
svr	-	+	~	+	~	-
rid	-	+	-	~	+	~
enc	-	+	~	-	+	~
mlp	-	+	~	+	~	~
dtr	-	+	-	+	+	-
rfr	-	+	~	+	+	~
knn	-	+	~	+	+	~

Table 5.1: Recommendation for the relationship algorithm-metric of the regressor. + (use-able for all data sets), $\sim$ (depends on data set), - (not use-able for all data sets).

Table 5.2 is the recommendation for the classification task. Here the results of seven data sets are summed up. The same criteria for +, $\sim$ and $-$ are given. The Equation 5.1 is also the summation criteria for the classification. Only six out of ninety algorithm-metric relations are not rated with +. ABC-hamming, -log, and -zero, SVM-prec and MLP-rec are all based on three negative p-value criteria. MLP-prec is grouped by four negative p-value criteria.

recommendation	acc	f1	fb	hamming	jaccard	log	prec	rec	zero
rfc	+	+	+	+	+	+	+	+	+
knn	+	+	+	+	+	+	+	+	+
svm	+	+	+	+	+	+	∼	+	+
dtc	+	+	+	+	+	+	+	+	+
gnb	+	+	+	+	+	+	+	+	+
lda	+	+	+	+	+	+	+	+	+
abc	+	+	+	∼	+	∼	+	+	∼
qda	+	+	+	+	+	+	+	+	+
mlp	+	+	+	+	+	+	-	∼	+
lrc	+	+	+	+	+	+	+	+	+

Table 5.2: Recommendation for the relation algorithm-metric of the classifier. + (use-able for all data sets), ∼ (depends on data set), - (not use-able for all data sets).

Data set C#1, **coarse grained soils**, trained as multi-class predictor shows for most of the algorithm-metrics good p-values and normalized standard deviation. The input features, different geological and geotechnical parameters, let the predictor perform well. Here all metrics can be recommended. The ABC algorithm shows for four metrics weak performance.

Data set C#2, **volcanic rocks**, with three geotechnical input parameters, show also good p-values and normalized standard deviation. Although fewer input parameters than in C#1, the performance is good. Also here the ABC-algorithm has no good performance for three metrics.

Data set C#3, **different rock types**, shows as C#2 with three geotechnical input parameters a stable data set. Beside the ABC-, also LRC-algorithm show for some metrics poorer performance. For the rest, all metrics perform well and are recommendable.

Data set C#4, **metamrophic and igneauos rocks**, with three geophysical-geotechnical inputs, a weaker performance for the rock type classification. For most algorithms, acc is not recommendable. Same for hamming and zero. MLP is for metrics not recommendable. Also ABC shows weak performance for most metrics. In general, all KNN-metrics and LRC-metrics can be recommended. Also the f1-metric, fb-metric, jaccard-metric and rec-metric are highly recommendable for this kind of data set.

Data set C#5, **different soils**, with two blow count inputs, a very good performance for most of the algorithm. MLP shows over-all metrics, beside log, a weak performance and is not recommendable. Also SVM is for most of the metrics not recommendable.

Data set C#6, **different soils**, with the input of the SPT parameters, shows high performance over most of the metrics. Only prec is moderate recommendable. The rest of the metrics perform well.

Data set C#7, **fine grained soils**, with six geotechnical inputs and the Atterberg classification as classifier output, shows for RFC, KNN and DTC for all metrics no recommendation. For the MLP, prec and rec is not useful. The rest of the algorithm-metric relation is highly recommendable.

Data set R#1, **undrained finish clays**, with six geotechnical inputs and the undrained shear strength as output of the regressor. The mer metric is, beside MLP-mer, not at all recommendable based on the p-value. DTR-mee and -r2, RFR-mse and MLP-r2 are also based on the negative p-value criteria. All other algorithm-metrics are recommendable.

Data set R#2, **different soils**, is the same data set as C#6. Here the regressor predicts the shear wave velocity by the SPT parameters. All of the mer, beside the DTR RFR, and the DTR-mee do not reach the p-value criteria. The DTR-r2 does not reach the s_n value. All other algorithm-metric relations are recommendable.

Data set R#3, **mixed stone**, is the same as the data set C#4. Geotechnical and geophysical input parameters are the basics for the prediction of the UCS. All mse show negative s_n criteria, SVR and RFR based on negative p-value criteria. Also all r2 metrics are based on negative p-value criteria or show no calculated values. The mer metric is for SVR, ENC, DTR, RFR and KNN not recommendable. The ENC-msle and MLP-mee show also negative s_n criteria, SVR-mee show no results. All other algorithm-metrics are recommendable.

Data set R#4, **mixed stone**, with input parameters of different geotechnical parameters, the predictor calculates the vertical strain. Here all mer show negeative p-value criteria and LIR-r2, SVR-r2 and DTR-r2 are negative by s_n criteria. So the data set has a high quality for recommendation.

Data set R#5, **different soil**, with input parameters of the piezocene testing indices, the predictor calculates the resilient modulus, which describes the elasticity modulus for quick/short applied loads. Beside ENC, DTR and RFR, all mer are negative p-values. Also LIR-, RID- and ENC-msle are based on negative p-value. The LAS-msle shows negative s_n criteria. All other algorithm-metrics are recommendable.

Data set R#6, **clays**, with four geotechnical input parameters and the liquid limit as the predictor output. Again mer shows for all algorithms negative p-value. The DTR-mer and KNN-mer are also with negative p-value.

Data set R#7, **volcanic rocks**, with two geotechnical input parameters and the UCS as the predictor output. Here only the mer algorithms, beside the RFR, show negative p-values.

Data set R#8, **coarse grained soils**, with four input parameters of the SPT test and the porosity for the predictor output. Here all algorithm-metrics show negative p-values, so the data set is not use-able.

Data set R#9, **soft finish clays**, with seven input parameters of the oedometer test and the void ratio as the predictor output. Here mer shows negative p-value criteria. The msle and r2 of LIR, LAS, RID, ENC and RFR have negative p-value. Also the LAS-, ENC-, RFR-mse and DTR-r2 show the same behaviour. The LIR, SVR, RID and DTR-mse have negative s_n criteria. These algorithm-metrics are not recommendable at all. The mar and mee metrics are for this data set over-all algorithms recommendable.

6 Conclusion

In summary most of the algorithm-metric relations are recommendable. For different data sets, this recommendation varies strongly. Repeated training and testing of the algorithms with a finite number of data sets gives a first idea, how future recommendations of the algorithm-metric recommendation can look like. In the sklearn package, a lot of other metrics and algorithm are available. Here more investigation must be done. Also with more different data sets, one can give a better and more detailed recommendation. The question is, if the scientific community is interested in such recommendations. Most supervisors use standard metrics and algorithm. If there is a need, the first step should be to check, which algorithm-metrics are mostly used by the geotechnical/geological community. One problem is, that in this field the ML is a young science and not widely known or applied. So investigation for recommendation of algorithm-metric relations will grow step by step with the growing knowledge and application of this scientific domain. Another future investigation could be the impact of the size of the data sets, especially for classification. Imbalanced classes can be a problem for training the algorithm and so the recommendation of the algorithm-metric could be not precise enough. In the end, all statements in chapter 5 are recommendations with a subjective influence, although trying to minimize these impacts. So by furthermore investigation, it could be that there will be never a clear decision criteria for choosing the right algorithm-metric pair for ones special data set case.

7 Bibliography

[1] Cpt geotechnik premstaller from tu graz homepage. https://www.tugraz.at/en/institutes/ibg/research/computational-geotechnics-group/database/. Accessed: 2022-11-04.

[2] fitter-package. https://fitter.readthedocs.io/en/latest/. Accessed: 2022-11-17.

[3] pandas - python data analysis library, accessed: 22.06.2022. URL: https://pandas.pydata.org/.

[4] Welcome to python.org, accessed: 22.06.2022. URL: https://www.python.org/.

[5] Alexei Botchkarev. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019. URL: https://doi.org/10.28945%2F4184, doi:10.28945/4184.

[6] Jianye Ching, Kuang-Hao Li, Kok-Kwang Phoon, and Meng-Chia Weng. Generic transformation models for some intact rock properties. *Canadian Geotechnical Journal*, 55(12):1702–1741, 2018. arXiv:https://doi.org/10.1139/cgj-2017-0537, doi:10.1139/cgj-2017-0537.

[7] Jianye Ching, Guan-Hong Lin, Jie-Ru Chen, and Kok-Kwang Phoon. Transformation models for effective friction angle and relative density calibrated based on generic database of coarse-grained soils. *Canadian Geotechnical Journal*, 54, 10 2016. doi:10.1139/cgj-2016-0318.

[8] Jianye Ching, Kok-Kwang Phoon, and Chih-Hao Chen. Modeling piezocone cone penetration (cptu) parameters of clays as a multivariate normal distribution. *Canadian Geotechnical Journal*, 51, 01 2014. doi:10.1139/cgj-2012-0259.

[9] Jianye Ching, Kok-Kwang Phoon, Yuan-Hsun Ho, and Meng-Chia Weng. Quasi-site-specific prediction for deformation modulus of rock mass. *Canadian Geotechnical Journal*, 58, 08 2020. doi:10.1139/cgj-2020-0168.

[10] Luis Cuadros-Rodríguez, Estefanía Pérez-Castaño, and Cristina Ruiz-Samblás. Quality performance metrics in multivariate classification methods for qualitative analysis. *TrAC Trends in Analytical Chemistry*, 80:612–624, 2016. URL: https://www.sciencedirect.com/science/article/pii/S016599361630084X, doi:https://doi.org/10.1016/j.trac.2016.04.021.

[11] Marco D'Ignazio, Kok-Kwang Phoon, Siew Ann Tan, and Tim Tapani Länsivaara. Correlations for undrained shear strength of finnish soft clays. *Canadian Geotechnical Journal*, 53(10):1628–1645, 2016. arXiv:https://doi.org/10.1139/cgj-2016-0037, doi:10.1139/cgj-2016-0037.

[12] Shuyin Feng and Paul J. Vardanega. A database of saturated hydraulic conductivity of fine-grained soils: probability density functions. *Georisk: Assessment and Management of Risk*

for Engineered Systems and Geohazards, 13(4):255–261, 2019. arXiv:https://doi.org/10.1080/17499518.2019.1652919, doi:10.1080/17499518.2019.1652919.

[13] Michael Heap and Marie E.S. Violay. The mechanical behaviour and failure modes of volcanic rocks: a review. *Bulletin of Volcanology*, 83(5):33, May 2021. URL: https://hal.archives-ouvertes.fr/hal-03547456, doi:10.1007/s00445-021-01447-2.

[14] Frank J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1951.10500769, arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1951.10500769, doi:10.1080/01621459.1951.10500769.

[15] L. Kingma and L.P. Mackay. *Futurism: Cartoons from Tomorrow: A Futuristic Comic Collection*. Running Press, 2019. URL: https://books.google.at/books?id=7yyDDwAAQBAJ.

[16] Songyu Liu, Haifeng Zou, Guojun Cai, Tejo Vikash Bheemasetti, Anand J. Puppala, and Jun Lin. Multivariate correlation among resilient modulus and cone penetration test parameters of cohesive subgrade soils. *Engineering Geology*, 209:128–142, 2016. URL: https://www.sciencedirect.com/science/article/pii/S0013795216301648, doi:https://doi.org/10.1016/j.enggeo.2016.05.018.

[17] Monica Susanne Löfman and Leena Katariina Korkiala-Tanttu. Transformation models for the compressibility properties of finnish clays using a multivariate database. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 16(2):330–346, 2022. arXiv:https://doi.org/10.1080/17499518.2020.1864410, doi:10.1080/17499518.2020.1864410.

[18] Monica Susanne Löfman and Leena Katariina Korkiala-Tanttu. Transformation models for the compressibility properties of finnish clays using a multivariate database. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 16(2):330–346, 2022. arXiv:https://doi.org/10.1080/17499518.2020.1864410, doi:10.1080/17499518.2020.1864410.

[19] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook, 2nd ed.* 01 2010.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL: http://jmlr.org/papers/v12/pedregosa11a.html.

[21] Guangren Shi. Chapter 1 - introduction. In Guangren Shi, editor, *Data Mining and Knowledge Discovery for Geoscientists*, pages 1–22. Elsevier, Oxford, 2014. URL: https://www.sciencedirect.com/science/article/pii/B9780124104372000011, doi:https://doi.org/10.1016/B978-0-12-410437-2.00001-1.

[22] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing Management*, 45(4):427–437, 2009. URL:

https://www.sciencedirect.com/science/article/pii/S0306457309000259, doi:https://doi.org/10.1016/j.ipm.2009.03.002.

[23] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.

[24] Shihao Xiao , Jie Zhang, Jiaming Ye, and Jianguo Zheng. Establishing region-specific n – vs relationships through hierarchical bayesian modeling. *Engineering Geology*, 287:106105, 03 2021. doi:10.1016/j.enggeo.2021.106105.

[25] Pufeng Du Yasen Jiao. Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quantitative Biology*, 4(4):320, 2016. URL: https://journal.hep.com.cn/qb/EN/abstract/article_18373.shtml, doi:10.1007/s40484-016-0081-2.

[26] Dongming Zhang, Yelu Zhou, Kok-Kwang Phoon, and Hongwei Huang. Multivariate probability distribution of shanghai clay properties. *Engineering Geology*, 273:105675, 2020. URL: https://www.sciencedirect.com/science/article/pii/S0013795219315868, doi:https://doi.org/10.1016/j.enggeo.2020.105675.

A Appendix

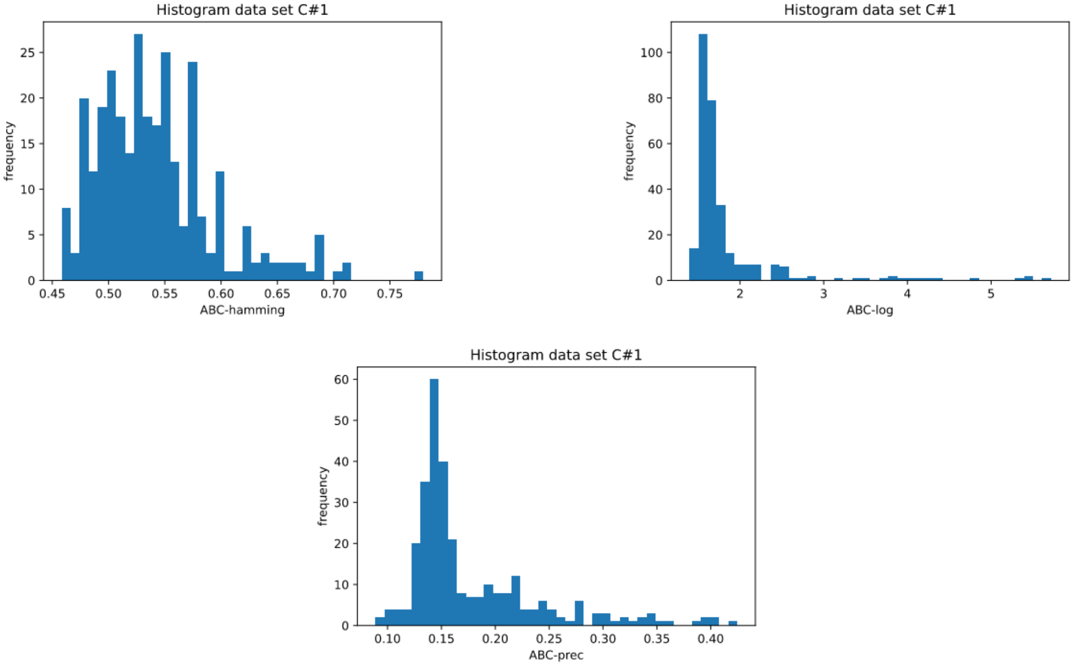


Figure A.1: Left top: Histogram of the ABC-hamming-loss; Right top: Histogram of the ABC-log-loss; Middle bottom: Histogram of the ABC-precision; related to Table 4.1 and data set C#1.

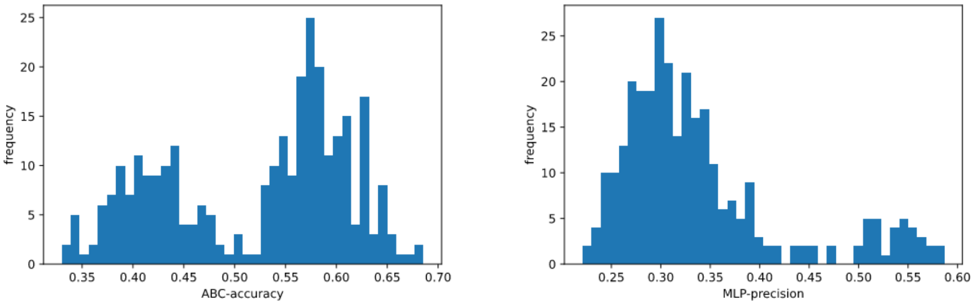


Figure A.2: Left: Histogram of the ABC-accuracy; Right: Histogram of the MLP-precision; related to Table 4.2 and data set C#2.

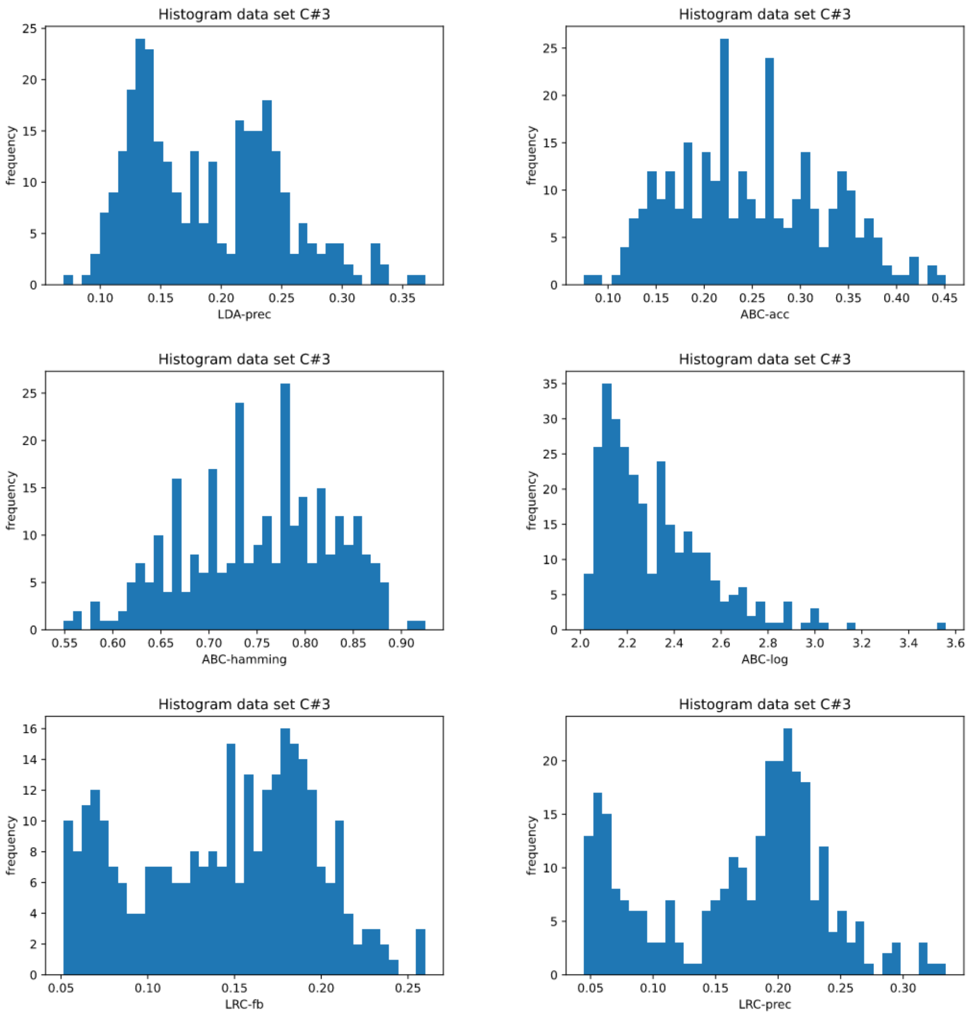
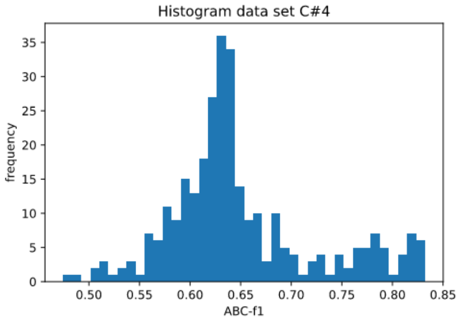
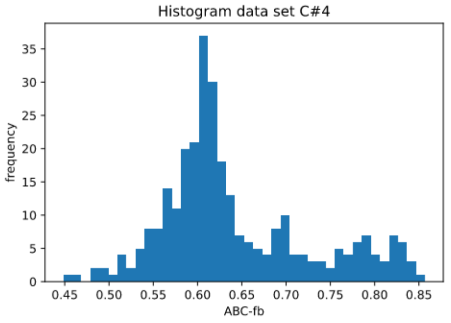
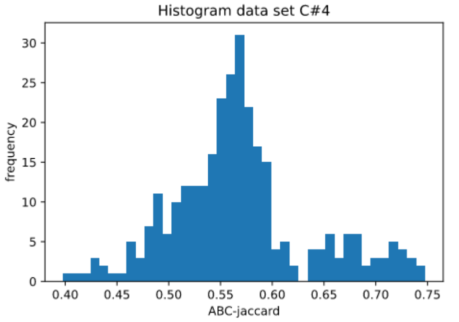
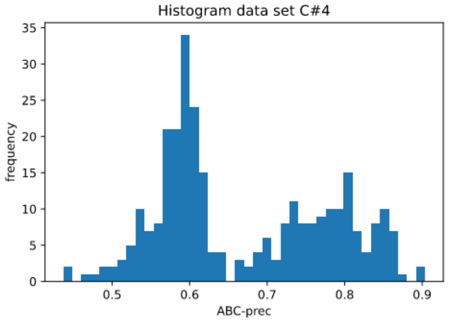


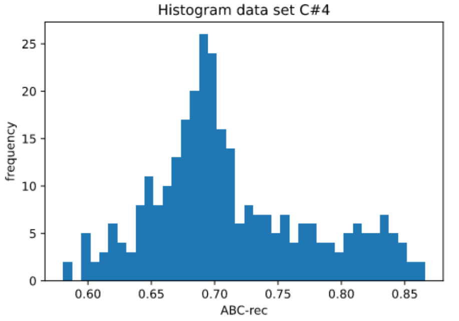
Figure A.3: Left top: Histogram of the LDA-precision; Right top: Histogram of the ABC-accuracy; Left middle: Histogram of the ABC-hamming-loss; Right middle: Histogram of the ABC-log-loss; Left bottom: Histogram of the LRC-fb-score; Right bottom: Histogram of the LRC-precision; related to Table 4.3 and data set C#3.

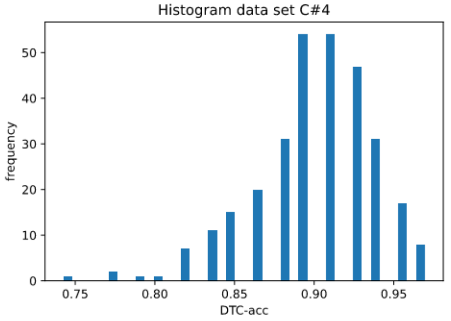

Histogram data set C#4

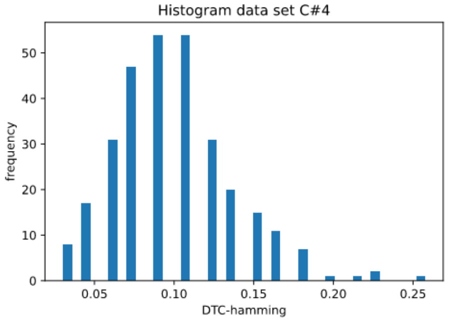

Histogram data set C#4

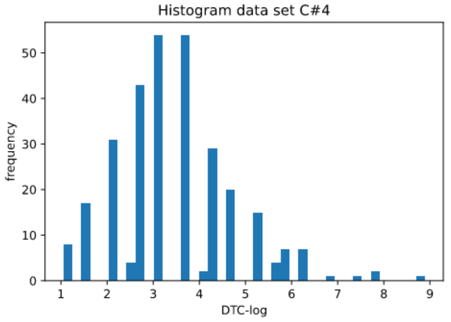

Histogram data set C#4

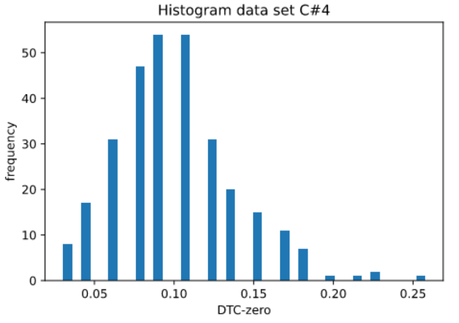

Histogram data set C#4

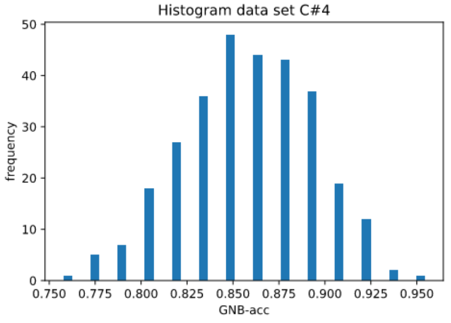

Histogram data set C#4

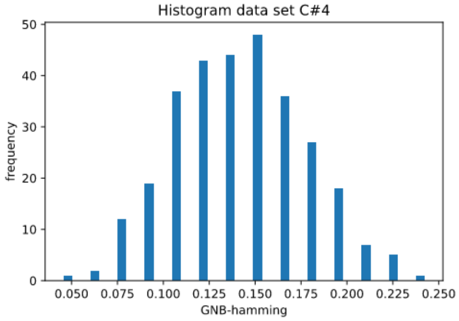

Histogram data set C#4

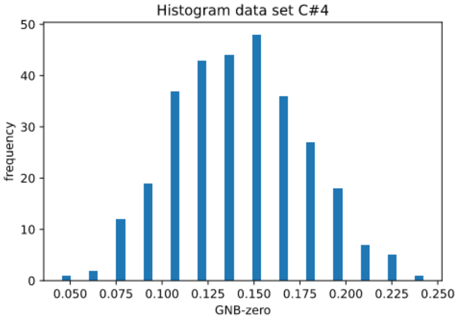

Histogram data set C#4

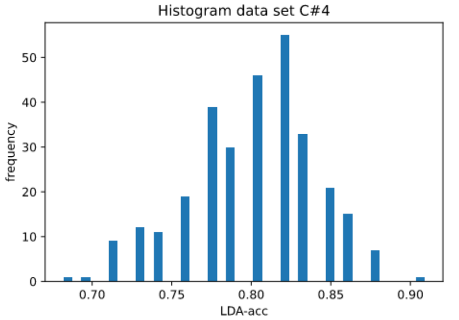

Histogram data set C#4

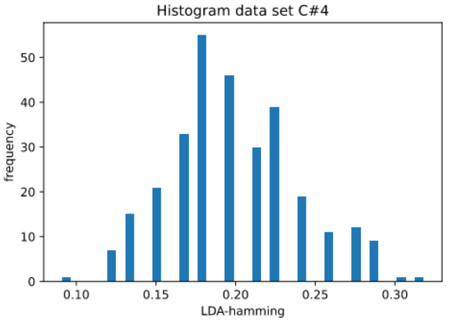

Histogram data set C#4

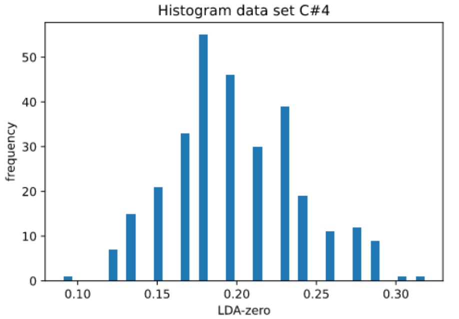

Histogram data set C#4

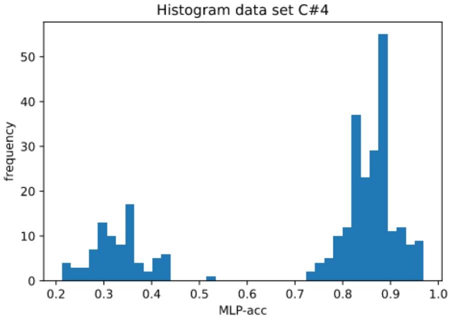

Histogram data set C#4
frequency
0
10
20
30
40
50
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0.225
0.250
GNB-hamming

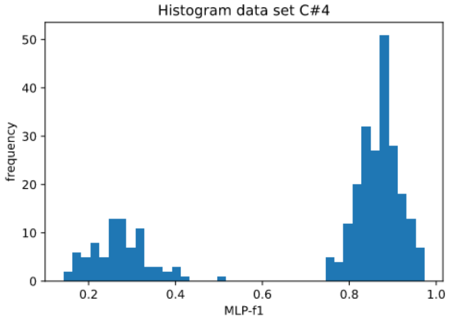

Histogram data set C#4
frequency
0
10
20
30
40
50
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0.225
0.250
GNB-zero

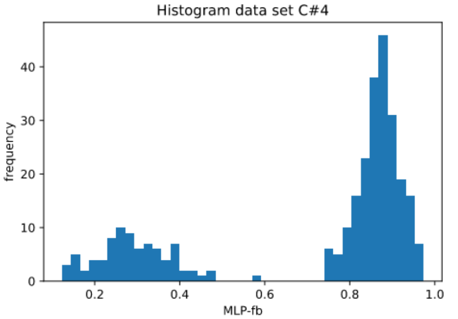

Histogram data set C#4
frequency
0
10
20
30
40
50
0.70
0.75
0.80
0.85
0.90
LDA-acc

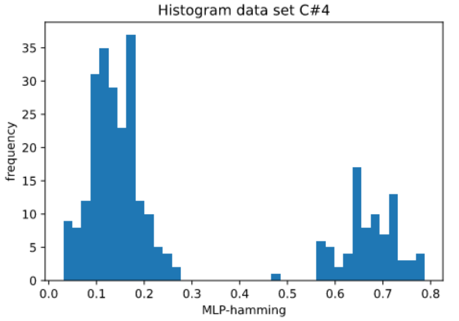

Histogram data set C#4
frequency
0
10
20
30
40
50
0.10
0.15
0.20
0.25
0.30
LDA-hamming

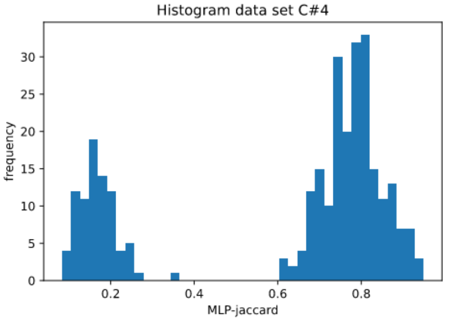

Histogram data set C#4
frequency
0
10
20
30
40
50
0.10
0.15
0.20
0.25
0.30
LDA-zero

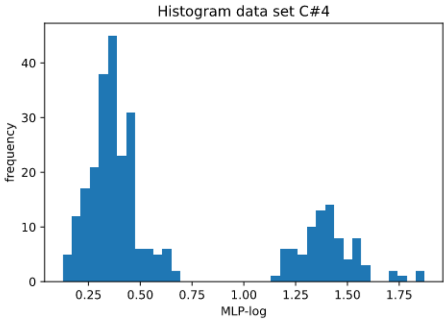

Histogram data set C#4
frequency
0
10
20
30
40
50
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
MLP-acc

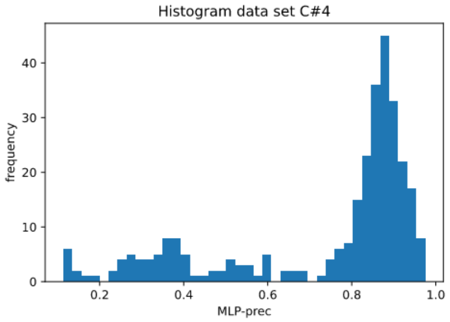

Histogram data set C#4
frequency
0
10
20
30
40
50
0.2
0.4
0.6
0.8
1.0
MLP-f1

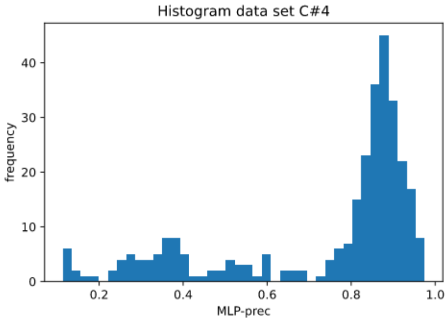

Histogram data set C#4
frequency
0
10
20
30
40
0.2
0.4
0.6
0.8
1.0
MLP-fb

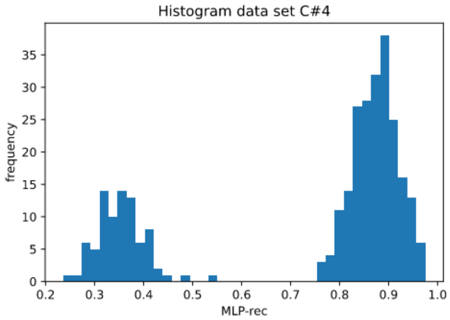

Histogram data set C#4
frequency
0
5
10
15
20
25
30
35
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
MLP-hamming

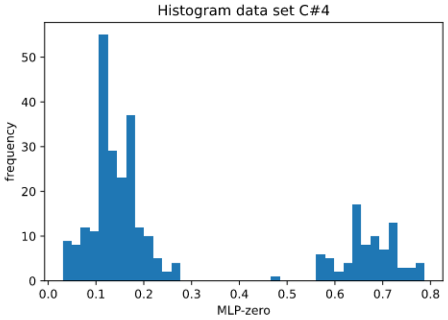

Histogram data set C#4
frequency
0
5
10
15
20
25
30
0.2
0.4
0.6
0.8
MLP-jaccard

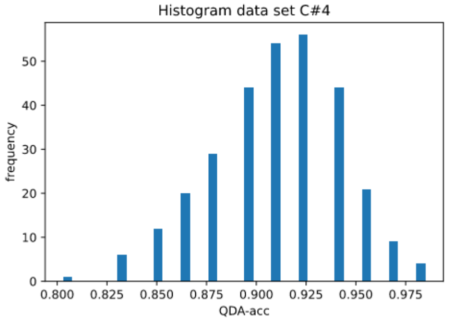

Histogram data set C#4
frequency
0
10
20
30
40
0.25
0.50
0.75
1.00
1.25
1.50
1.75
MLP-log

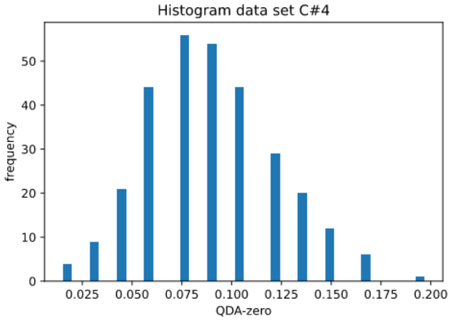

Histogram data set C#4
frequency
0
10
20
30
40
0.2
0.4
0.6
0.8
1.0
MLP-prec

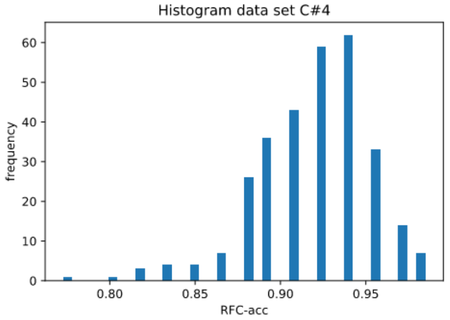

Histogram data set C#4
frequency
0
10
20
30
40
0.2
0.4
0.6
0.8
1.0
MLP-prec

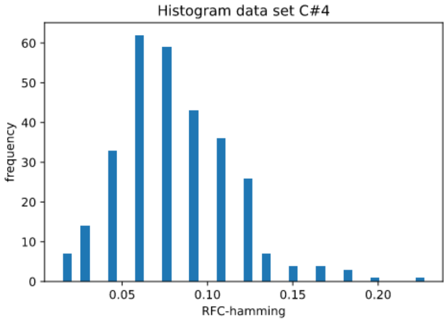

Histogram data set C#4
frequency
0
5
10
15
20
25
30
35
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
MLP-rec

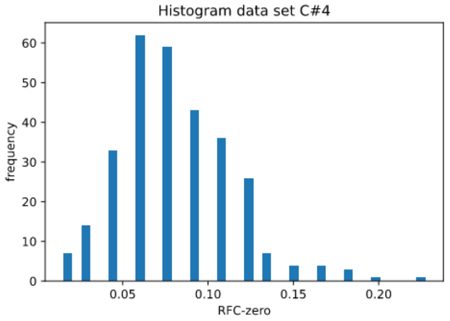

Histogram data set C#4
frequency
0
10
20
30
40
50
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
MLP-zero


Histogram data set C#4
frequency
0
10
20
30
40
50
0.800
0.825
0.850
0.875
0.900
0.925
0.950
0.975
QDA-acc


Histogram data set C#4
frequency
0
10
20
30
40
50
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
QDA-zero


Histogram data set C#4
frequency
0
10
20
30
40
50
60
0.80
0.85
0.90
0.95
RFC-acc


Histogram data set C#4
frequency
0
10
20
30
40
50
60
0.05
0.10
0.15
0.20
RFC-hamming


Histogram data set C#4
frequency
0
10
20
30
40
50
60
0.05
0.10
0.15
0.20
RFC-zero

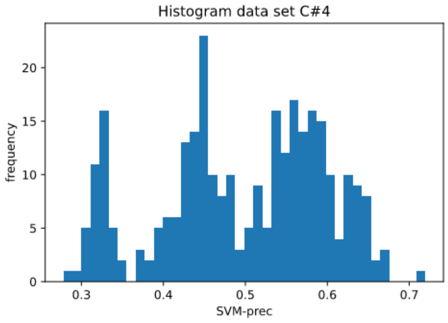
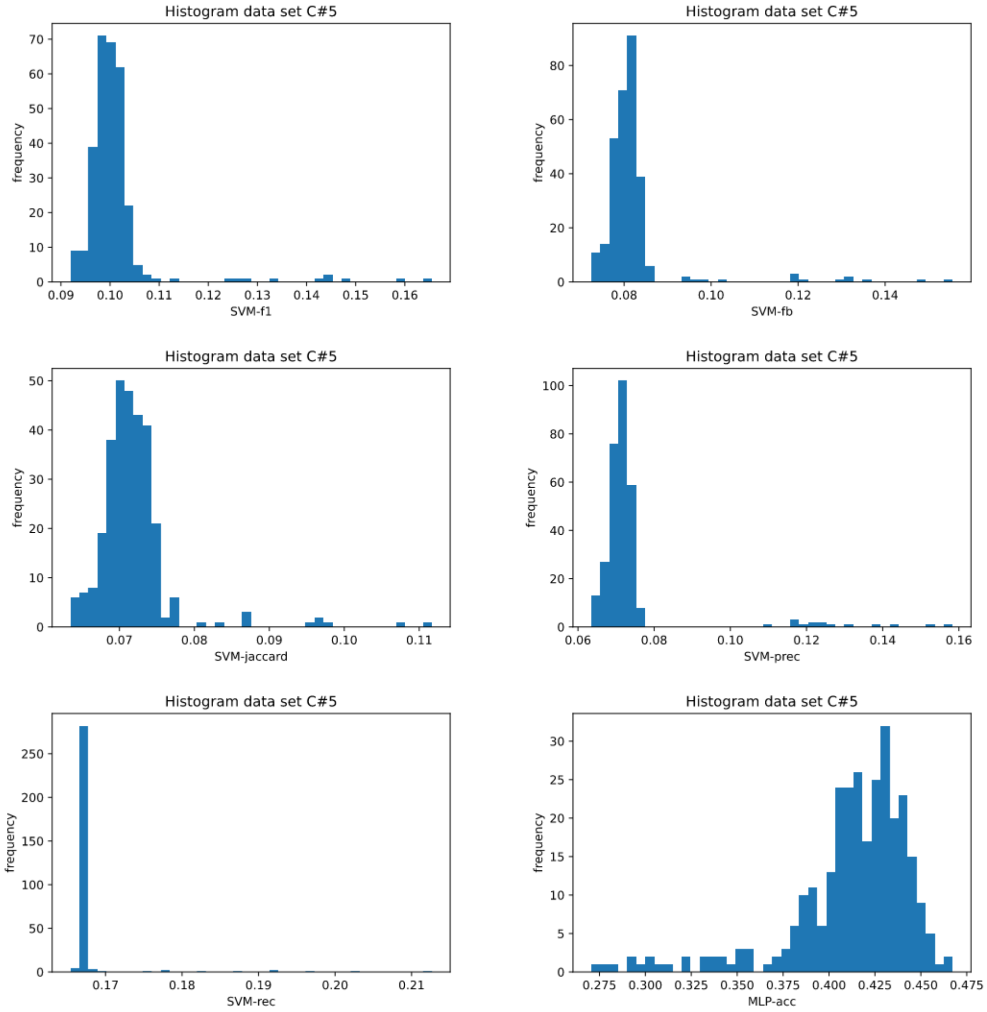


Figure A.4: Histograms of the algorithm-metrics distribution, which don not reach the p-value criteria; related to Table 4.4 and data set C#4.



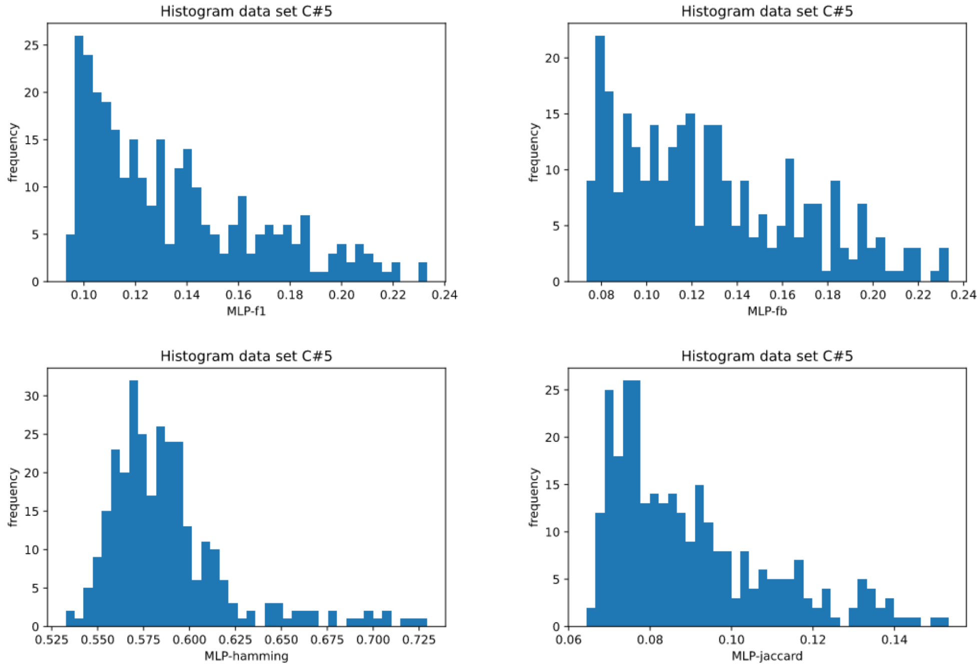


Figure A.5: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.5 and data set C#5.

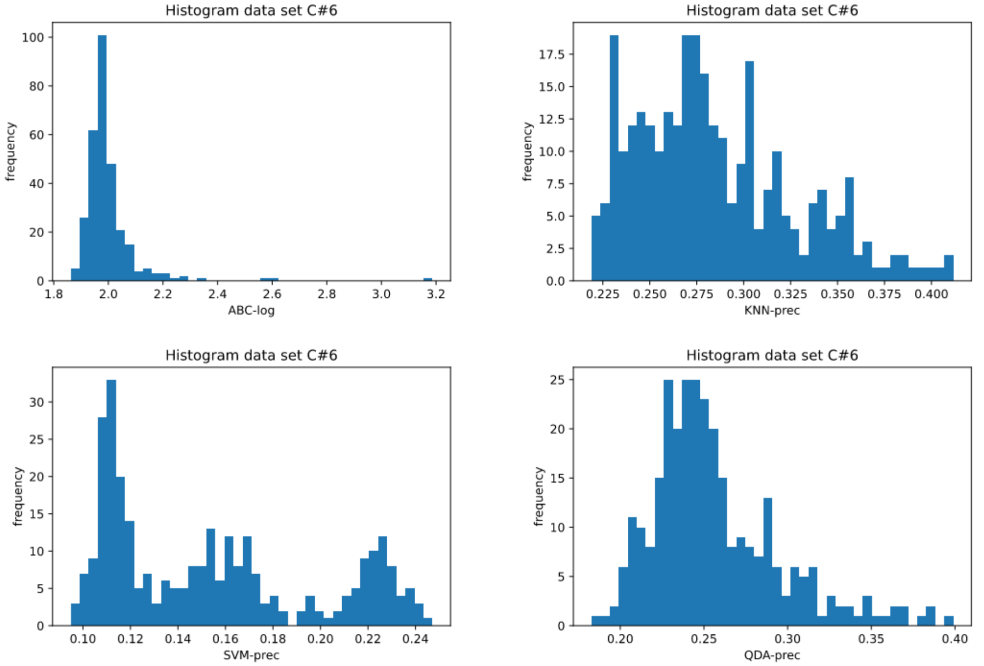
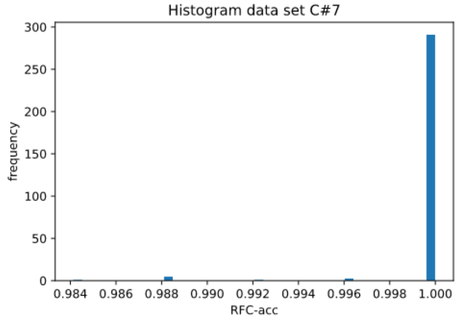
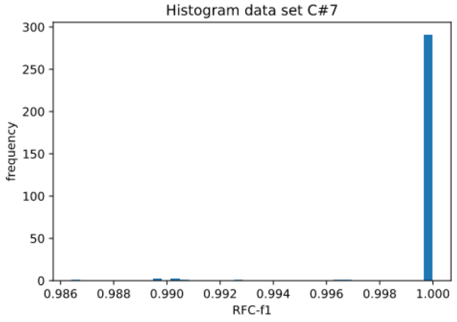
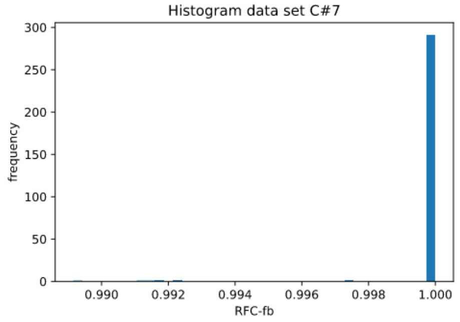
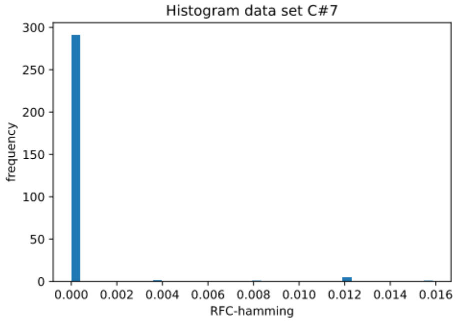


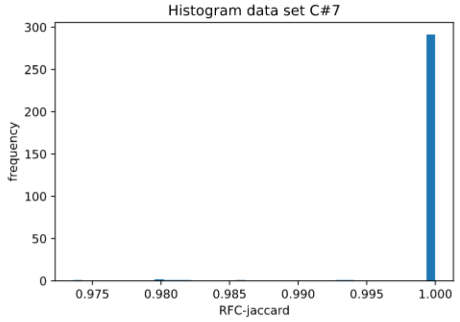
Figure A.6: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.6 and data set C#6.

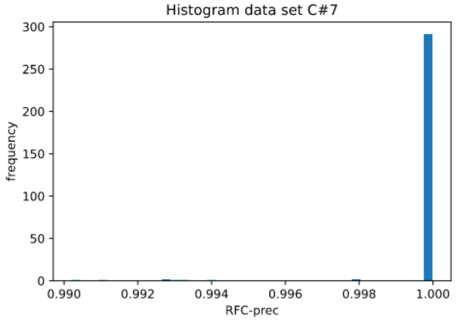

Histogram data set C#7

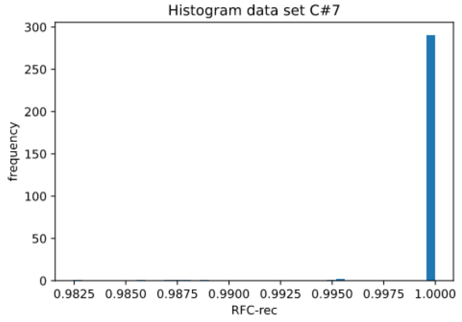

Histogram data set C#7

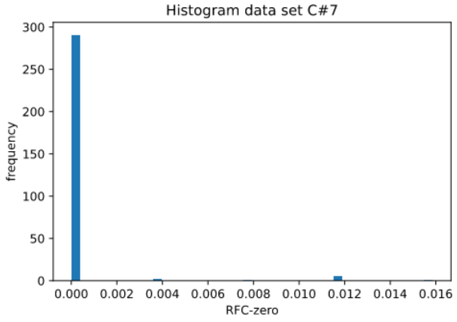

Histogram data set C#7

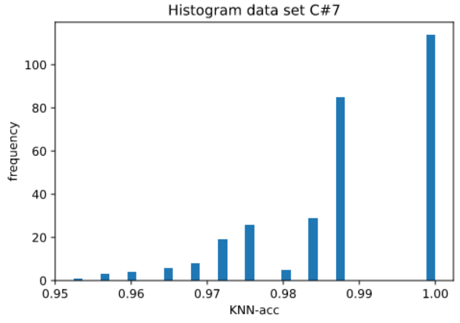

Histogram data set C#7

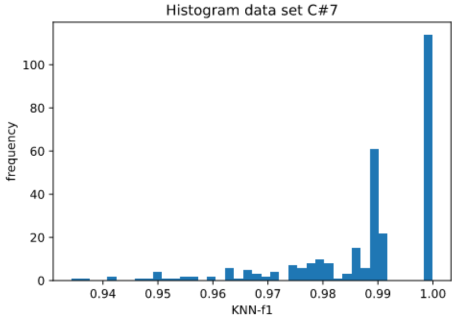

Histogram data set C#7


Histogram data set C#7

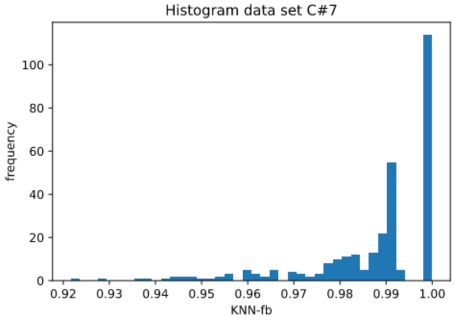

Histogram data set C#7


Histogram data set C#7

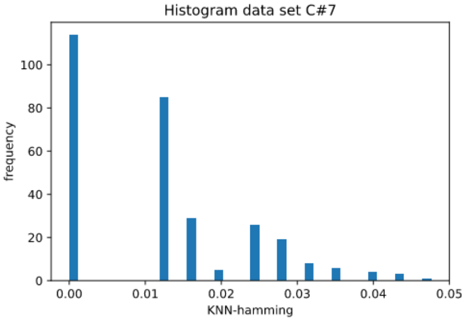

Histogram data set C#7


Histogram data set C#7

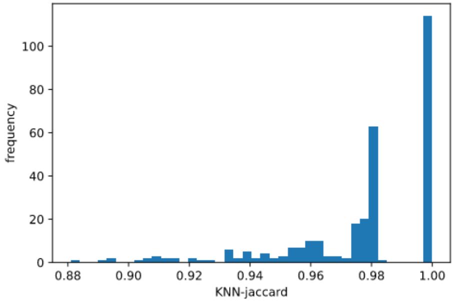
Histogram data set C#7



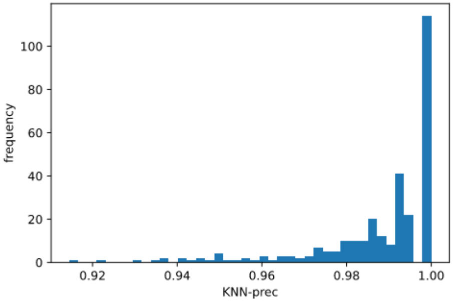
Histogram data set C#7



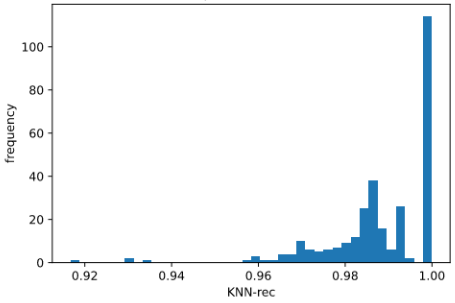
Histogram data set C#7



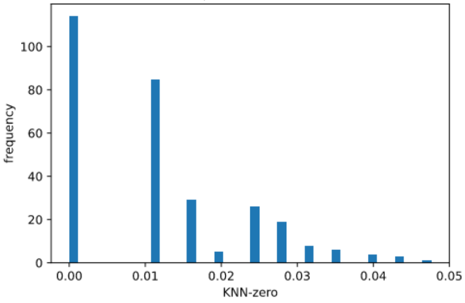
Histogram data set C#7



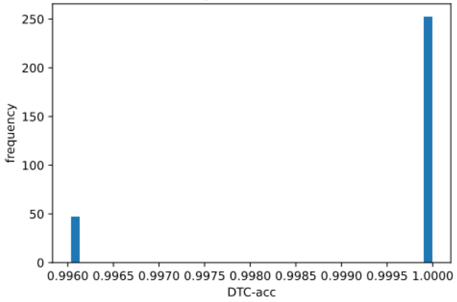
Histogram data set C#7



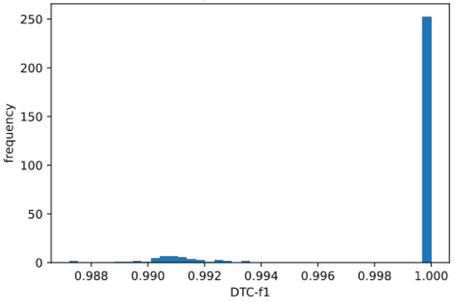
Histogram data set C#7



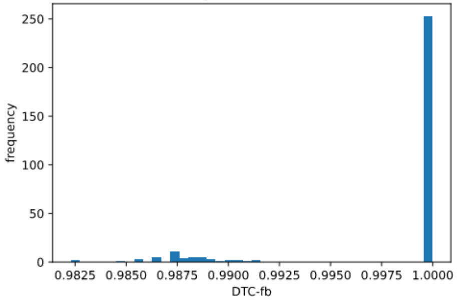
Histogram data set C#7



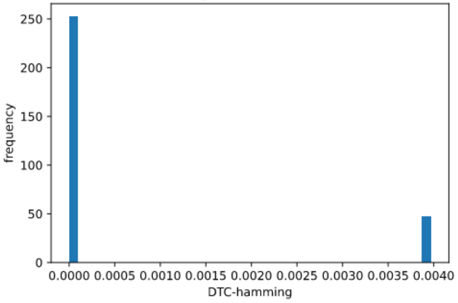
Histogram data set C#7



Histogram data set C#7



Histogram data set C#7



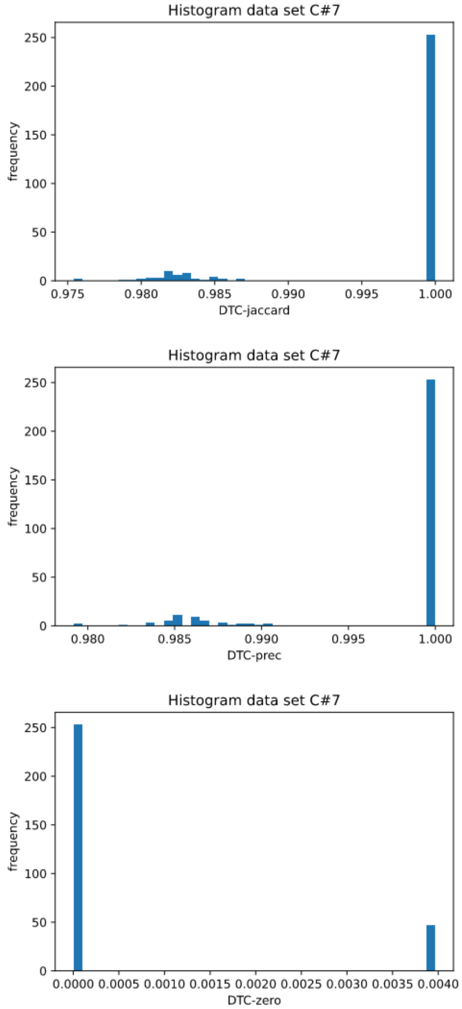
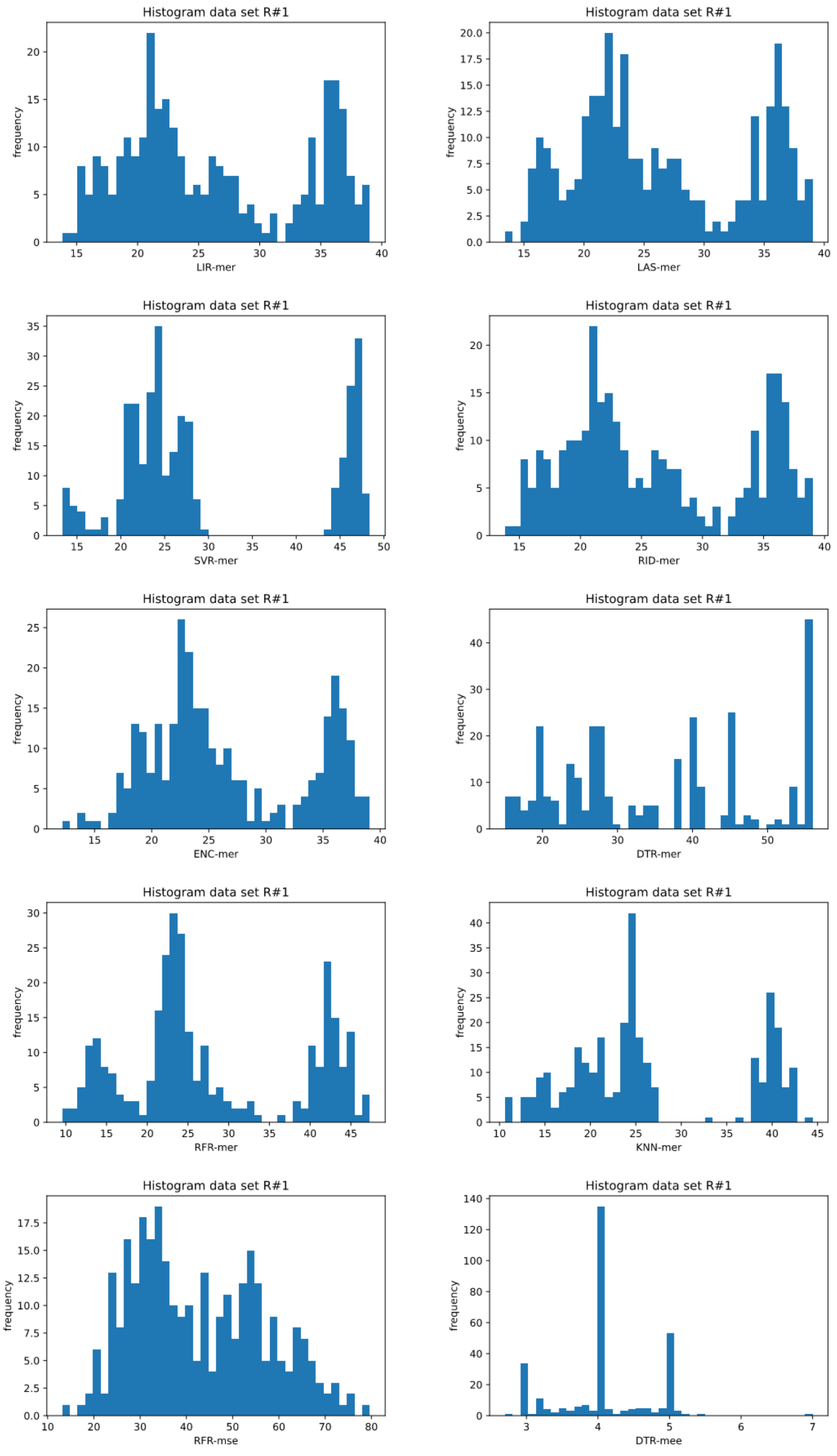


Figure A.7: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.7 and data set C#7.


Histogram data set R#1
frequency
0
5
10
15
20
15
20
25
30
35
40
LIR-mer
Histogram data set R#1
frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
15
20
25
30
35
40
LAS-mer
Histogram data set R#1
frequency
0
5
10
15
20
25
30
35
15
20
25
30
35
40
45
50
SVR-mer
Histogram data set R#1
frequency
0
5
10
15
20
15
20
25
30
35
40
RID-mer
Histogram data set R#1
frequency
0
5
10
15
20
25
15
20
25
30
35
40
ENC-mer
Histogram data set R#1
frequency
0
10
20
30
40
20
30
40
50
DTR-mer
Histogram data set R#1
frequency
0
5
10
15
20
25
30
10
15
20
25
30
35
40
45
RFR-mer
Histogram data set R#1
frequency
0
5
10
15
20
25
30
35
40
10
15
20
25
30
35
40
45
KNN-mer
Histogram data set R#1
frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
10
20
30
40
50
60
70
80
RFR-mse
Histogram data set R#1
frequency
0
20
40
60
80
100
120
140
3
4
5
6
7
DTR-mee

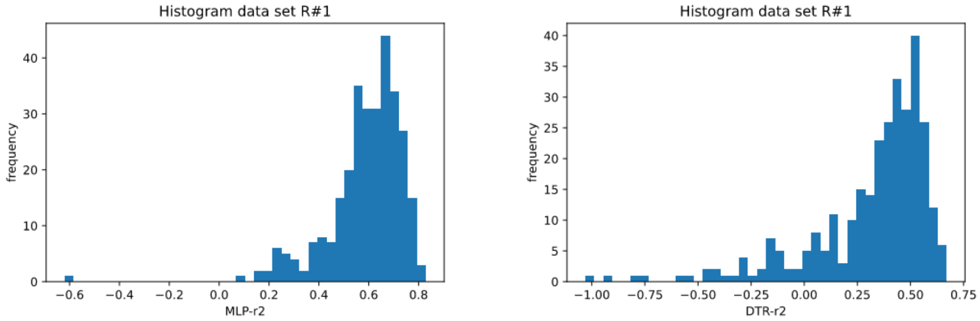
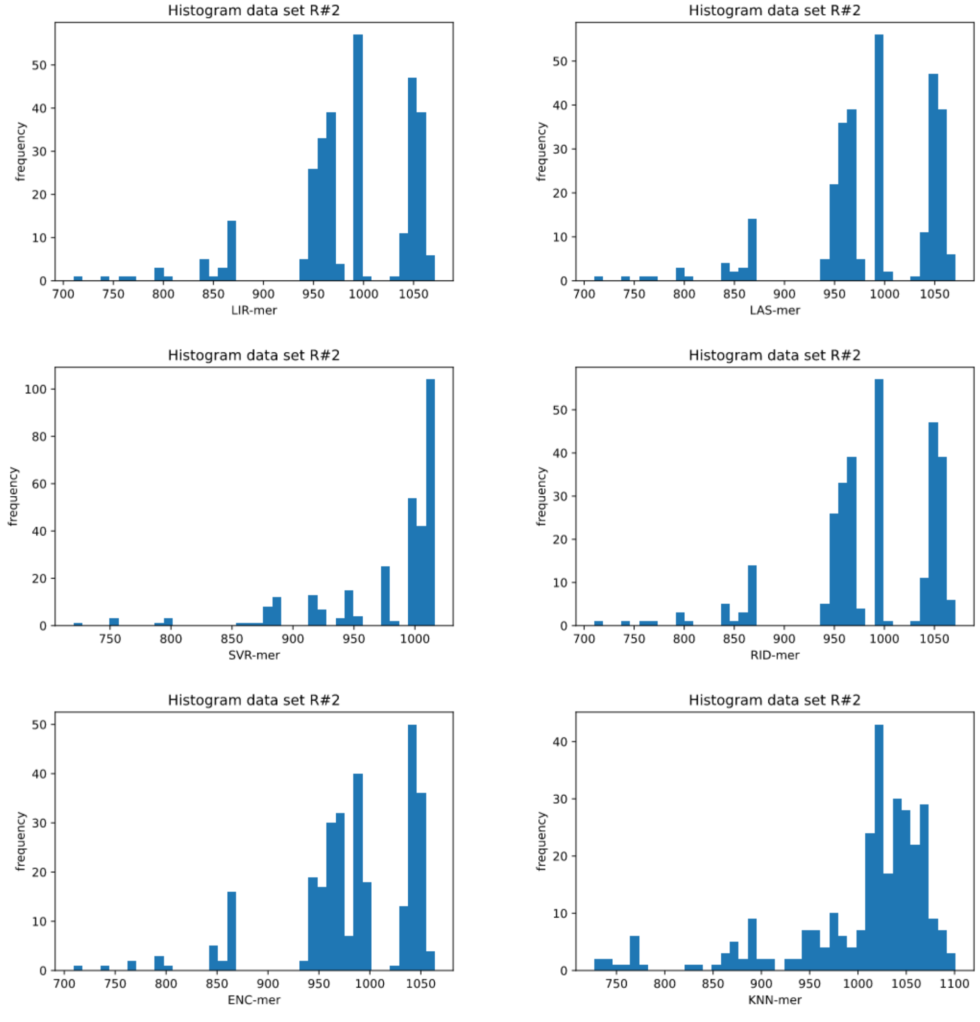


Figure A.8: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.8 and data set R#1.



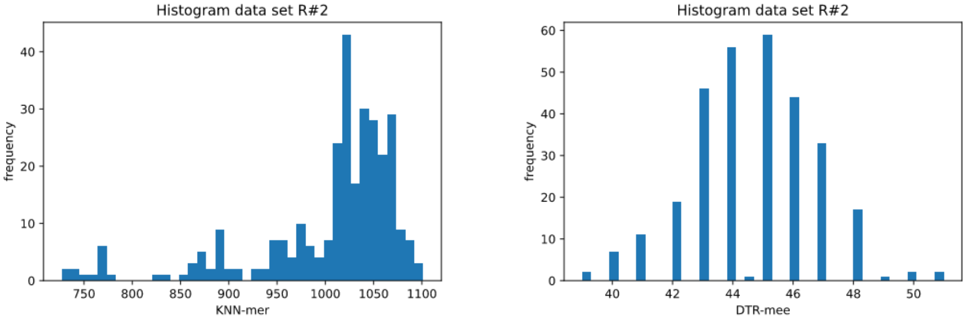
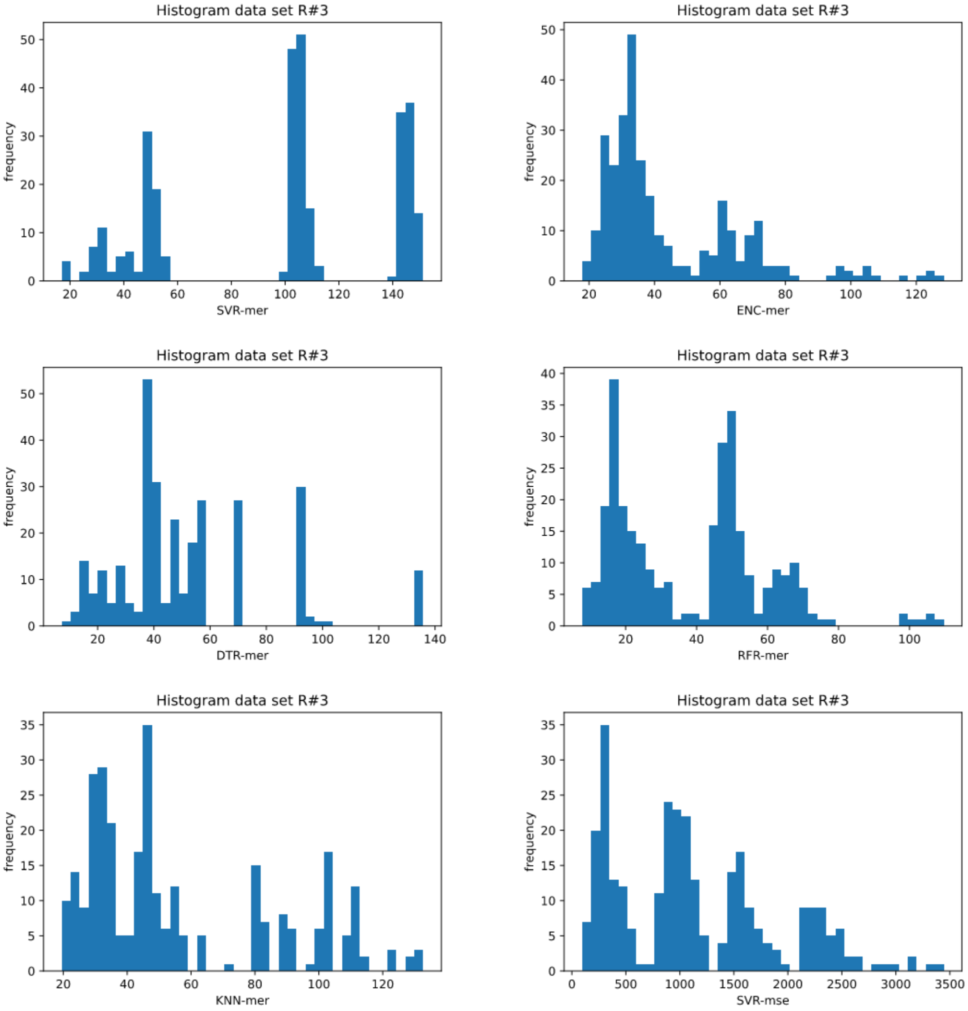


Figure A.9: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.9 and data set R#2.



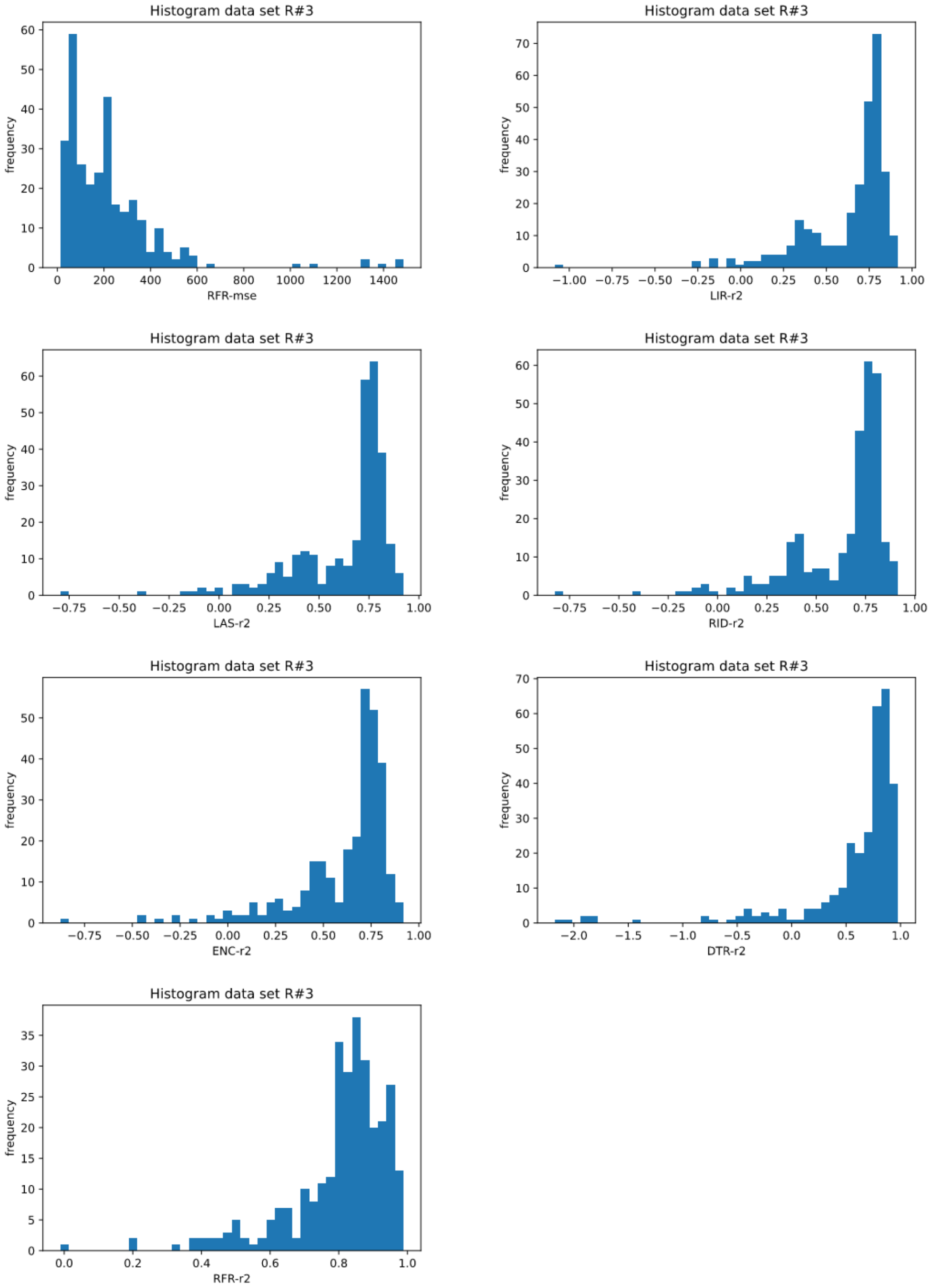
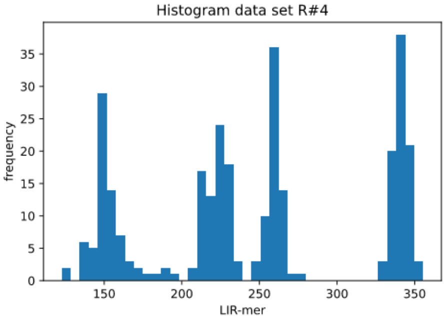
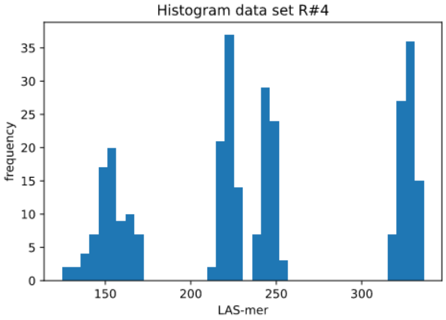
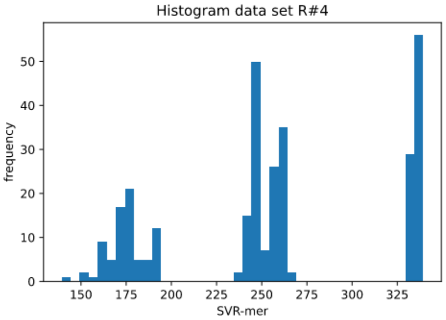
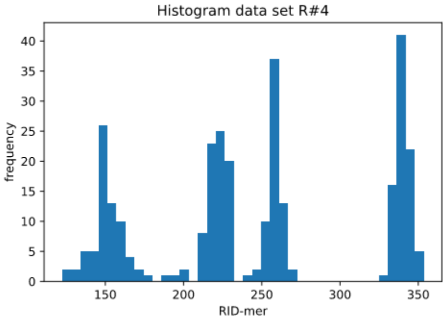


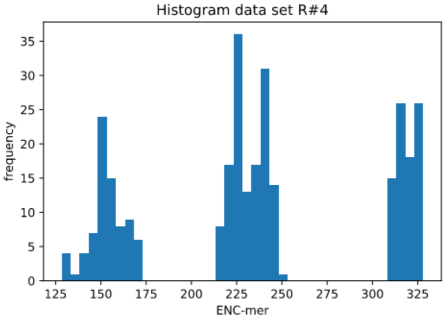
Figure A.10: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.10 and data set R#3.

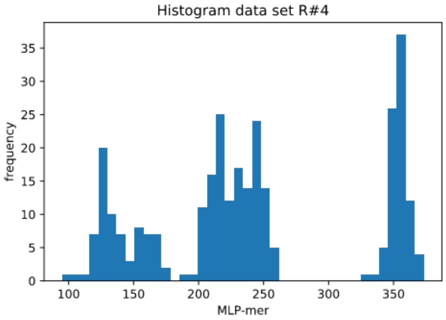

Histogram data set R#4
frequency
0
5
10
15
20
25
30
35
150
200
250
300
350
LIR-mer

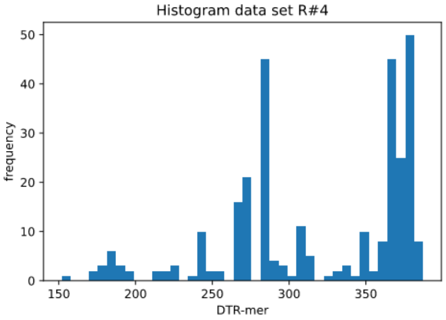

Histogram data set R#4
frequency
0
5
10
15
20
25
30
35
150
200
250
300
LAS-mer

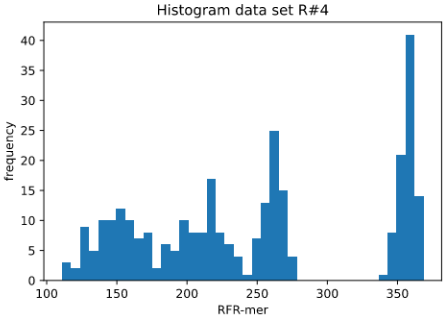

Histogram data set R#4
frequency
0
10
20
30
40
50
150
175
200
225
250
275
300
325
SVR-mer


Histogram data set R#4
frequency
0
5
10
15
20
25
30
35
40
150
200
250
300
350
RID-mer


Histogram data set R#4
frequency
0
5
10
15
20
25
30
35
125
150
175
200
225
250
275
300
325
ENC-mer


Histogram data set R#4
frequency
0
5
10
15
20
25
30
35
100
150
200
250
300
350
MLP-mer


Histogram data set R#4
frequency
0
10
20
30
40
50
150
200
250
300
350
DTR-mer


Histogram data set R#4
frequency
0
5
10
15
20
25
30
35
40
100
150
200
250
300
350
RFR-mer

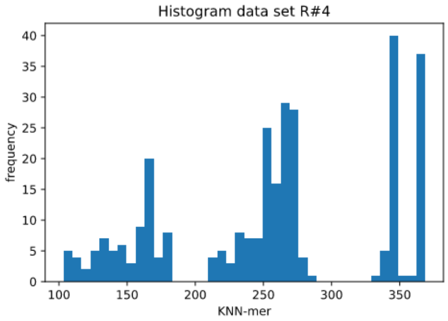
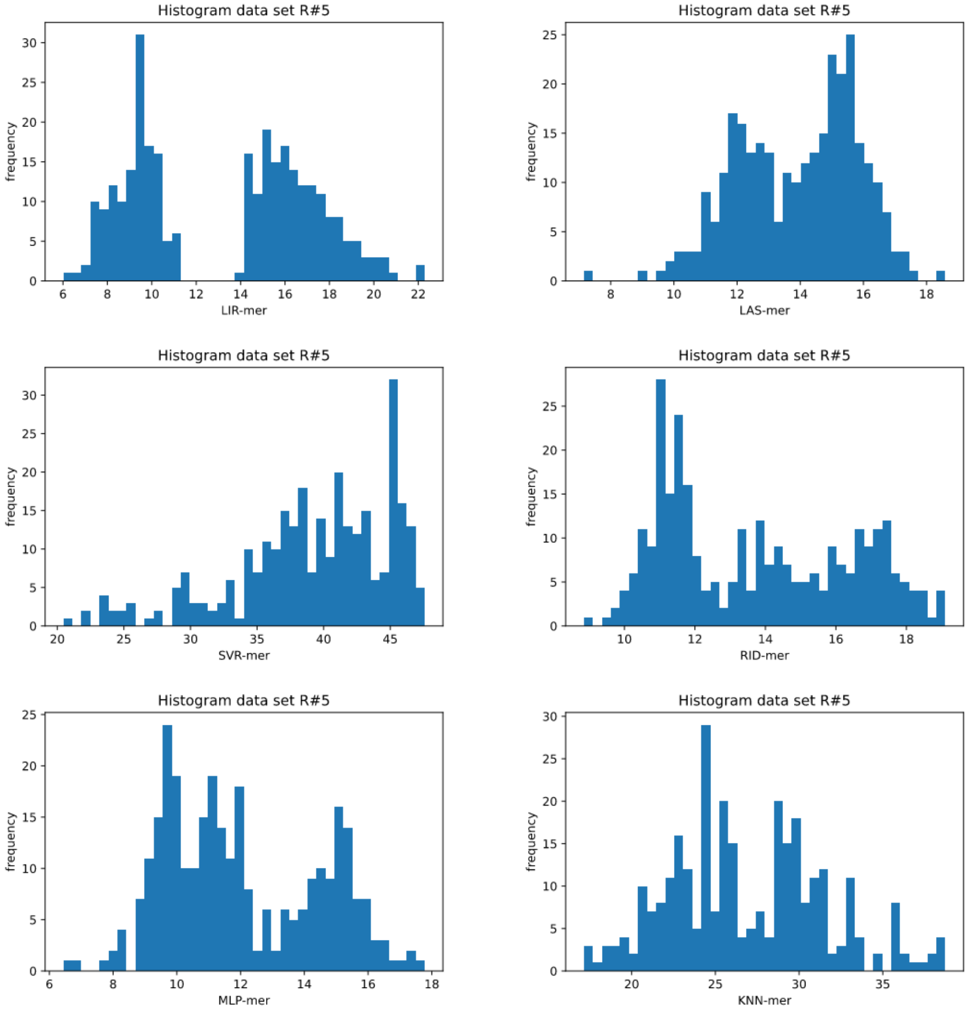


Figure A.11: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.11 and data set R#4.



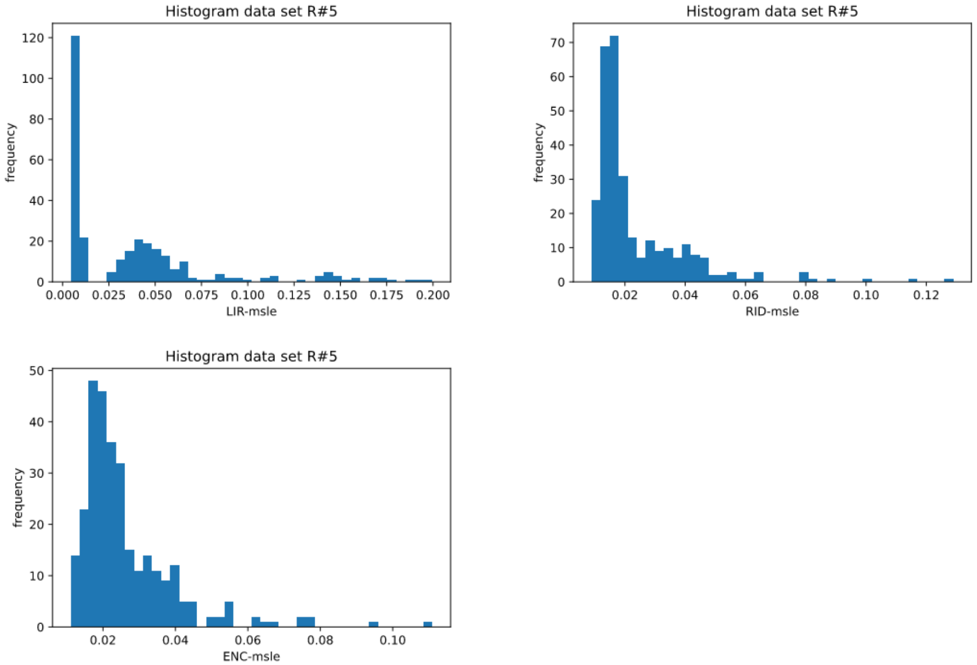
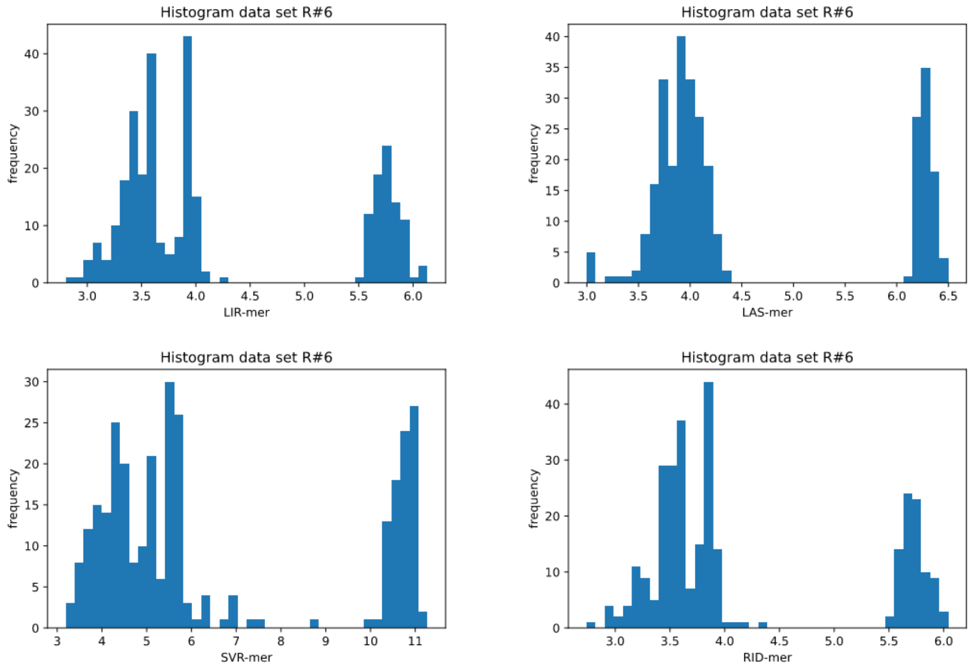


Figure A.12: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.12 and data set R#5.



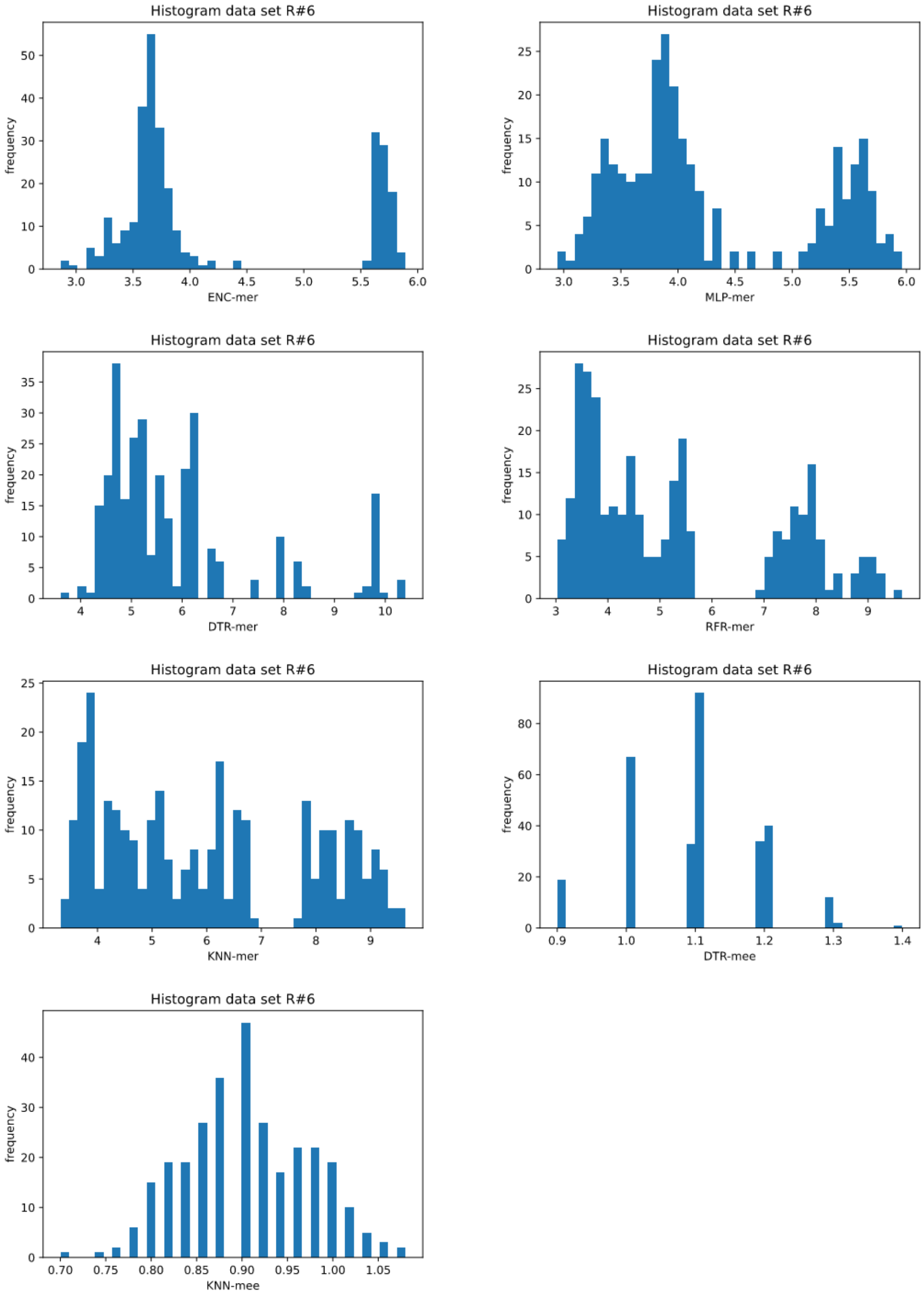


Figure A.13: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.13 and data set R#6.

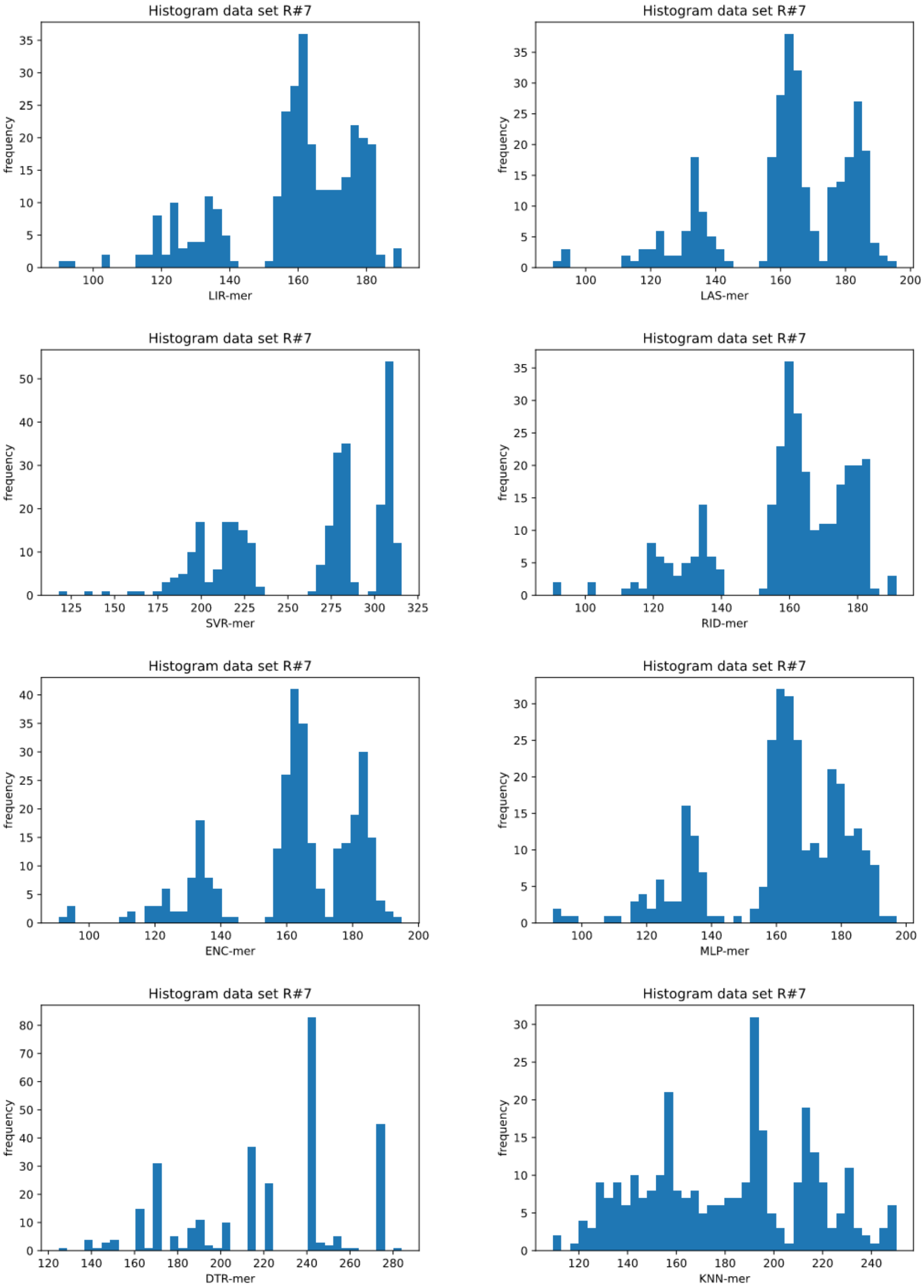
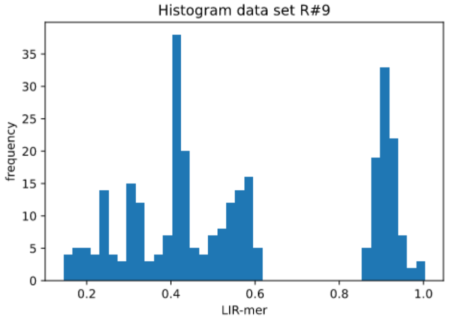
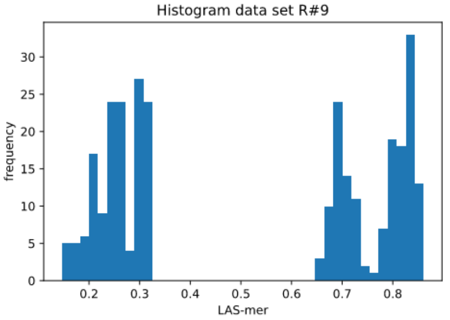
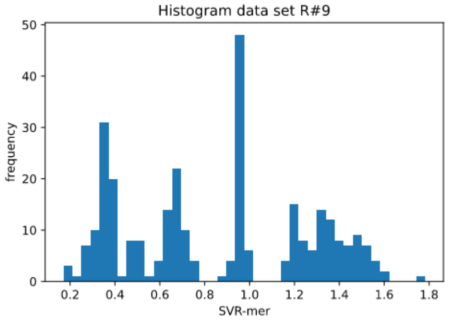
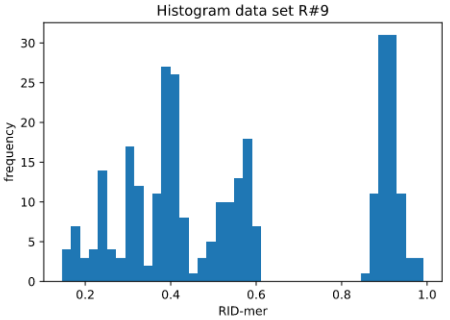


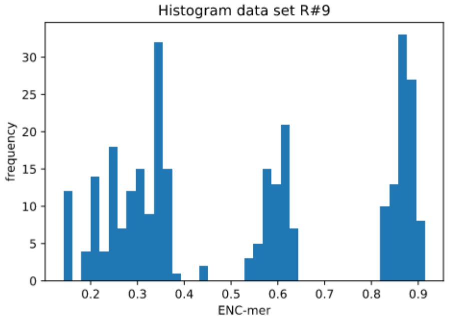
Figure A.14: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.14 and data set R#7.

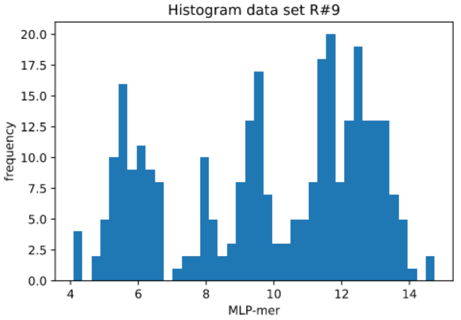

Histogram data set R#9

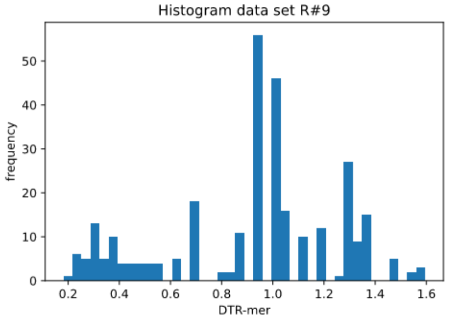

Histogram data set R#9

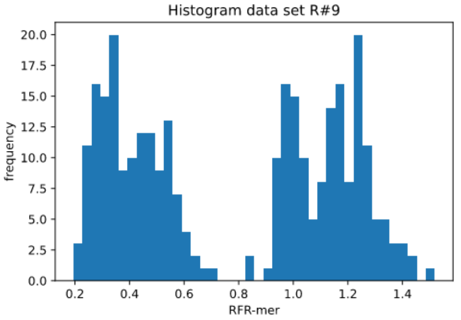

Histogram data set R#9

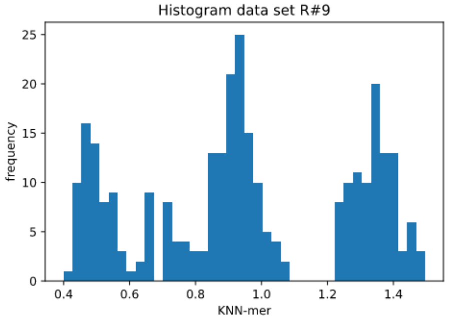

Histogram data set R#9

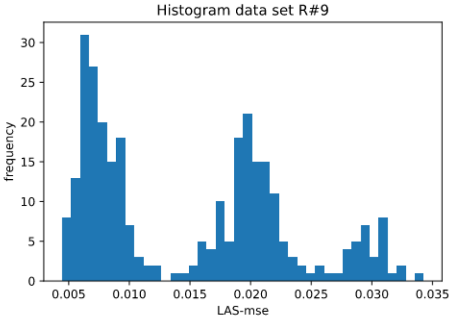

Histogram data set R#9

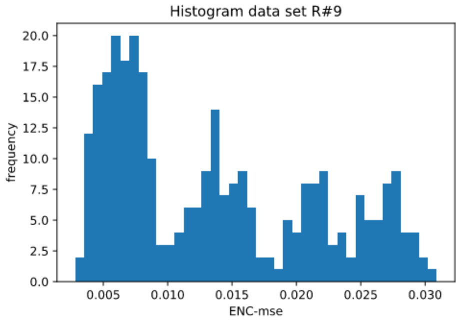

Histogram data set R#9

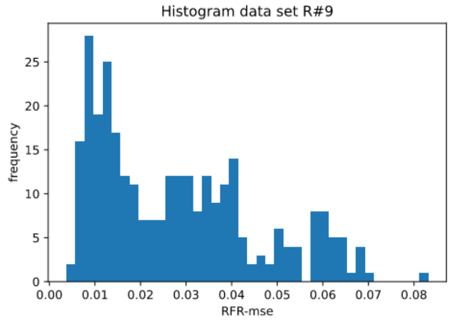

Histogram data set R#9

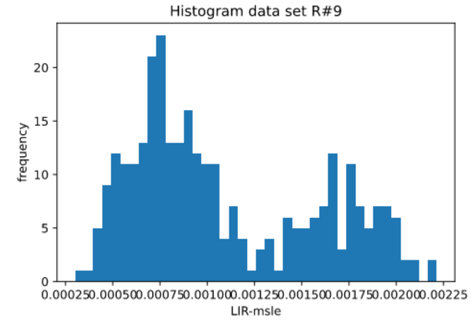

Histogram data set R#9

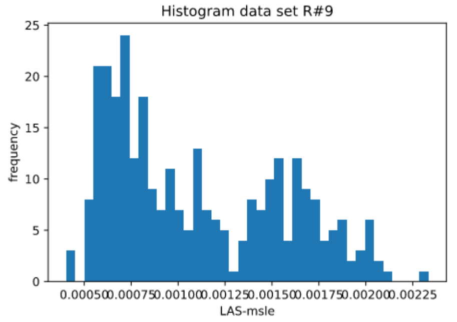

Histogram data set R#9

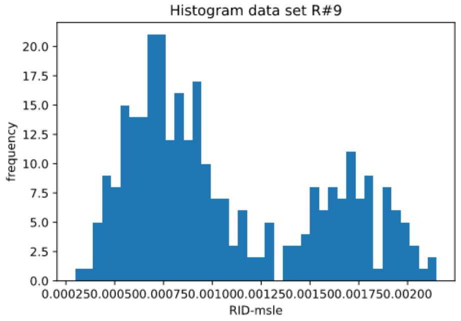

Histogram data set R#9

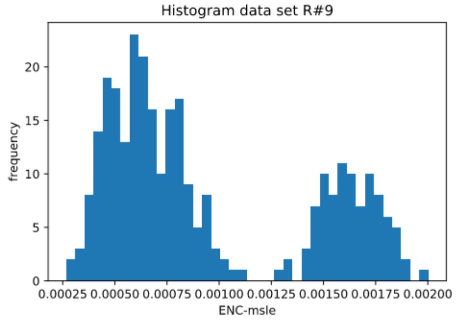

Histogram data set R#9
frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.005
0.010
0.015
0.020
0.025
0.030
ENC-mse

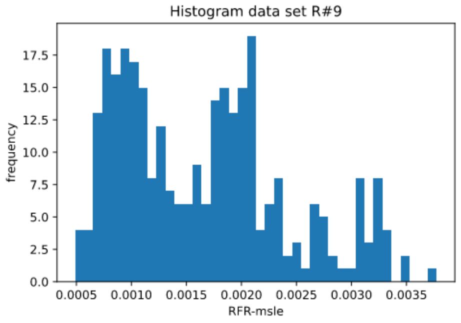

Histogram data set R#9
frequency
0
5
10
15
20
25
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
RFR-mse

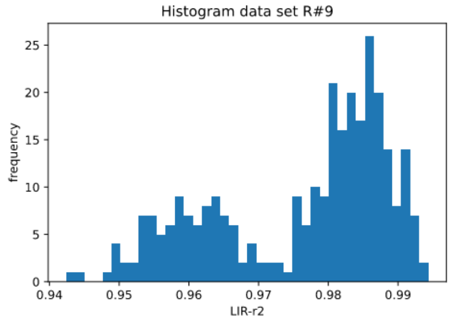

Histogram data set R#9
frequency
0
5
10
15
20
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0.00200
0.00225
LIR-msle

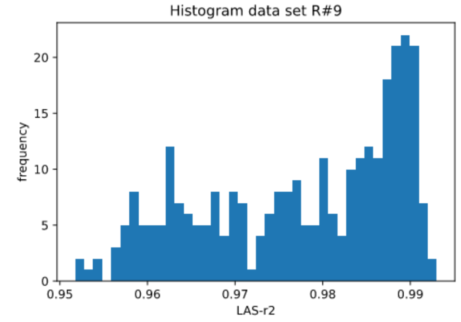

Histogram data set R#9
frequency
0
5
10
15
20
25
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0.00200
0.00225
LAS-msle

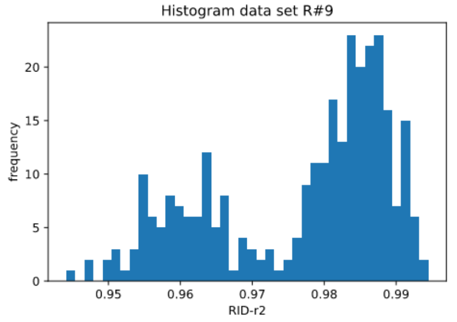

Histogram data set R#9
frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0.00200
RID-msle


Histogram data set R#9
frequency
0
5
10
15
20
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0.00200
ENC-msle


Histogram data set R#9
frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
0.0035
RFR-msle


Histogram data set R#9
frequency
0
5
10
15
20
25
0.94
0.95
0.96
0.97
0.98
0.99
LIR-r2


Histogram data set R#9
frequency
0
5
10
15
20
0.95
0.96
0.97
0.98
0.99
LAS-r2


Histogram data set R#9
frequency
0
5
10
15
20
0.95
0.96
0.97
0.98
0.99
RID-r2

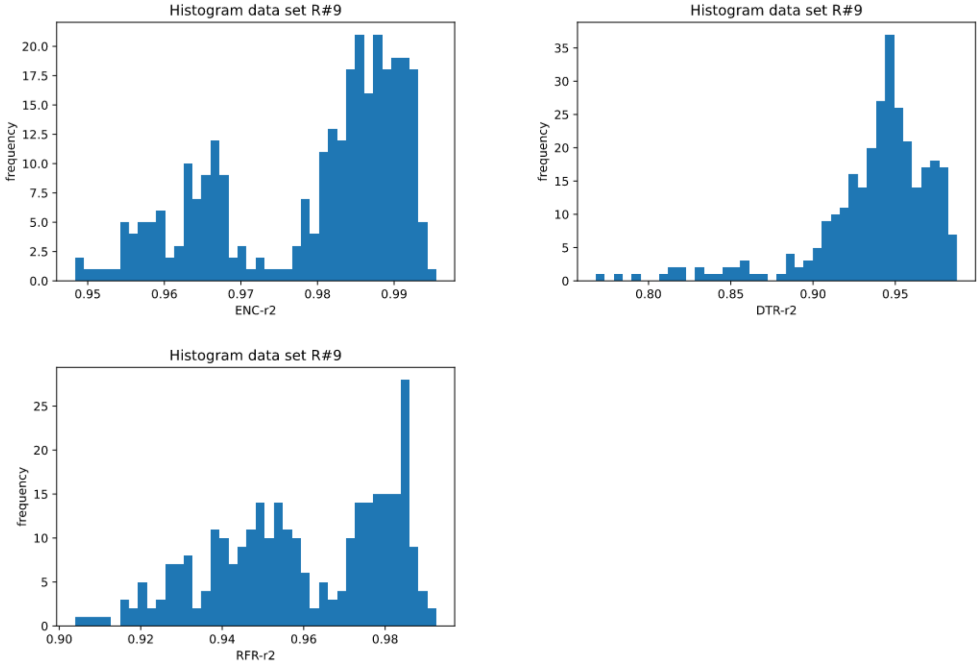


Figure A.15: Histograms of the algorithm-metrics distribution, which do not reach the p-value criteria; related to Table 4.16 and data set R#9.