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

May 6, 2023

Matthias Hahn*[1], Alla Sapronova[2] and Marlene Villeneuve[1]
[1]Chair of Subsurface Engineering, Monatnuniversität Leoben, Austria
[2]Institute of Rock Mechanics and Tunneling, TU Graz, Austria
*presenting author (email: mattthias.hahn@stud.unileoben.ac.at)

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. So for different algorithms one metric describe a comparable value. Most supervisors use common metrics, but which metric is use able for which algorithm with respect to geotechnic/geology. The following chapter discuss this relation for different algorithm and metrics considering a variation of data sets.

1 Introduction

Like all models, machine learning algorithms have errors and can not predict the output exact. To get an idea how well an applied algorithm works, performance measuring is needed [6]. This investigation focus on regressors and classifiers. An example for classification in geoscience is labeling rocks into lithologies, based on different geotechnical parameters. For regression, instead of labeling lithology, the predictor calculate the uniaxial compressive strength or the vertical tension. The metrics for this two tasks show different aims. Regressor metrics show the performance based on the distance between predicted and supervised output data [3]. Classifiers metrics compare, how many predicted values are in the right/positive class [9]. So different metrics are developed for comparing the algorithms. The question is: in geoscience, are all metrics useful and trust-able for all algorithms and data sets? If not, what are the limitations? The knowledge of this topic is little in the geoscience community [8]. The following paper is structured: First, we describe the used methodologies, predictors and metrics. Next we describe the used data sets and the preparation. Third we present and discuss the results for the used methodologies. Last a recommendation is given for the usefulness of the different algorithm-metric relations.

2 Methodology, Predictors, Metrics

2.1 Methodology

For giving a recommendation of an algorithm-metric relation, the idea is to train and measure the performance for several hundred similar data sets. So a statistical statement can be given for how stable the algorithm-metric perform. This repeated training and performance measuring fading out the subjective influence of the supervisor. Every algorithm and metric need input parameters, like the used solver. Many of these parameters have no strict criteria for use, so the experience of the supervisor is needed. In this case, for the same data set different metric results are possible and the recommendation for the algorithm-metric relation is subjective. For our investigation, the needed amount of eligible data sets is not given. Eligible data sets are sets, where the classes are not too imbalanced or the regression has only training data for a small intervall. Also comparable data sets, for example different SPT measurement for soft clays, are not available for this research. So the question rises, how to perform the training and performance measuring in respect for a statistical approach? The idea was to repeat the training and performance measuring several times. Every repeated training the data set is splitted randomly into training and testing part. This was done with the $train\_test\_split()$ command of the sklearn package[7] of python [2]. 30% was the test part of the data set. The training-testing loop was performed 300 times. The result is a distribution of how often a algorithm-metric result is calculated. Figure 1 shows the histogram of the K-Nearest Neighbor with the accuracy metric. It shows also the fitted probability distributions, this is explained later. How to evaluate the stability of the algorithm-metric relation? The distribution shows some kind of probability distribution. Every probability distribution can be described by different parameters like mean, median, standard deviation,... For evaluating the stability we decided to use the normalized standard deviation. A small standard deviation describes a distribution with less variation. The idea is, that such an algorithm-metric relation implicates stability for different data sets and the subjectivity of the supervisor parameters have only small impact. In the end, it would be use-able for the similar variation of data sets in geoscience. The package Fitter [1] fits the available probability distribution of SciPy [10] for a given distribution. Figure 1 shwos the fitted probability distributions by package Fitter. The distributions are ordered by the Kolmogorow-Smirnow-Test p-value. For a given hypothesis H_0 the KS-pvalue tells how good the data can be described by the probability distribution. If the p-value < 0.05, H_1 hypothesis is favorable, so the data can be described better by another distribution [5].

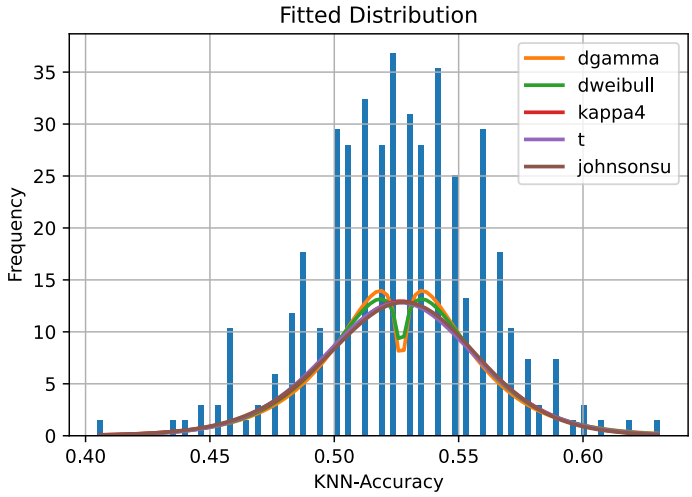

Figure 1: Histogram for the accuracy of KNN predictor

2.2 Predictors and Metrics

Table 1 summarize the used predictors and metrics for the regression. As few as possible algorithm parameters are used. The value for random_state = 42. For metrics no parameters are needed. Table 2 is the same like before, only for classifier.

short cut	predictor	parameters	short cut	metric
lir	LinearRegression()		mer	max_error()
las	Lasso()	random_state	mae	mean_absolute_error()
svr	svm.SVR()		mse	mean_squared_error()
rid	Ridge()	random_state	msle	mean_squared_log_error()
enc	ElasticNetCV()	recompute='auto', random_state	mee	median_absolute_error()
mlp	MLPRegressor()	max_iter = 2000, random_state	r2	r2_score()
dtr	DecisionTreeRegressor()	random_state		
rfr	RandomForestRegressor()	random_state		
knn	KNeighborsRegressor()			
gpr	GaussianProcessRegressor()	random_state		

Table 1: Regression predictors and metrics. Right side: Short cut, sklearn command of the predictor and the used parameters. Left side: short cut and sklearn command of the used metrics.

short cut	predictor	parameters	short cut	metric
rfc	RandomForestClassifier()	random_state	acc	accuracy_score()
knn	KNeighborsClassifier()		f1	f1_score()
svm	SVC()	random_state, probability=True	fb	fbeta_score()
dtc	DecisionTreeClassifier()		hamming	hamming_loss()
gnb	GaussianNB()		jaccard	jaccard_score()
lda	LinearDiscriminantAnalysis()		log	log_loss()
abc	AdaBoostClassifier()	random_state	prec	precision_score()
qda	QuadraticDiscriminantAnalysis()		rec	recall_score()
mlp	MLPClassifier()	random_state, max_iter = 4000	zero	zero_one_loss()
lrc	LogisticRegression()	random_state, solver='sag', max_iter=4000		

Table 2: Classification predictors and metrics. Right side: Short cut, sklearn command of the predictor and the used parameters. Left side: short cut and sklearn command of the used metrics.

3 Data and Results

The link to the Github server can be found here: https://github.com/rmttugraz/MatthiasHann.git

3.1 Data

16 data sets are used from the website of the TC304 Engineering Practice of Risk Assessment & Management of the International Society of Soil Mechanics and Geotechnical Engineering (ISSMGE). Finish soft clays, Cone Penetration Test, mixed rocks, CPT in clays and cohesive subgrade soils, volcanic rocks, Shanghai clays and coarse grained soils are the data sets used for regression. Here different features are used for predicting geotechnical parameters. Some of the regression data sets are also used for classification and visa versa. Here sandy/clayes soils, igneous-sedimentary-metamorphic rocks and finnish fine grained soils complete the data. Most of the predicted values are rock classes or geotechnical classes. Nine data sets for regression and seven data sets for classification result in 1170 algorithm-metric histograms. This histograms can be found at the Github server stated before. Also the description and data preparation of every data set can be found in the server.

3.2 Results

For every algorithm, one probability distribution over all metrics is chosen, so the metrics are comparable for one predictor. The result is for every data set a table with the normalized standard deviation s_n for each algorithm-metric relation. Table 3 shows an example of one of these tables, of normalized standard deviation for undrained finish clay [4]. The green values have a p-value $<$ 0.05. The red cell background show that the s_n is $>$ 0.5. Beside the mer, the results are based on positive p-value criteria. The only negative s_n criteria is not trustful because of negative p-value criteria. The recommendation of the data mainly based on the s_n criteria. This evaluation can be found for every s_n table at a document at the Github server.

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 3: The normalized standard s_n deviation for undrained finish clay [4]. Green: s_n with p-value<0.05. Red: s_n>0.5.

4 Interpretation and Recommendation

The results discussed in subsection 3.2 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 4 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 1 shows the summation criteria for the regressor. If the value reaches >30% the over-all recommendation is $\sim$. For values > 50% it is $-$.

$$Regressor_{crit} = \frac{\sum pvalue_{neg} + 2 * s_n(pvalue_{pos})}{n_{datasets}} * 100 \tag{1}$$

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 4: Recommendation for the relationship algorithm-metric of the regressor. + (use-able for all data sets), ∼ (depends on data set), - (not use-able for all data sets).

Table 5 is the recommendation for the classification task. Here the results of seven data sets are summed up. The same criteria for +, ∼ and − are given. The Equation 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: 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).

A recommendation for every data set can be found in the document at the Github server.

5 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 section 4 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.

6 References

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

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

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.