Metrics for Benchmarking and Uncertainty Quantification: Quality, Applicability, and a Path to Best Practices for Machine Learning in Chemistry

Gaurav Vishwakarma,† Aditya Sonpal, and Johannes Hachmann

1Department of Chemical and Biological Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260, United States
2Computational and Data-Enabled Science and Engineering Graduate Program, University at Buffalo, The State University of New York, Buffalo, NY 14260, United States
3New York State Center of Excellence in Materials Informatics, Buffalo, NY 14203, United States

This review aims to draw attention to two issues of concern when we set out to make machine learning work in the chemical and materials domain, i.e., statistical loss function metrics for the validation and benchmarking of data-derived models, and the uncertainty quantification of predictions made by them. They are often overlooked or underappreciated topics as chemists typically only have limited training in statistics. Aside from helping to assess the quality, reliability, and applicability of a given model, these metrics are also key to comparing the performance of different models and thus for developing guidelines and best practices for the successful application of machine learning in chemistry.

I. ASSESSING MACHINE LEARNING MODELS

The rapid advancement and transformation of machine learning (ML) technology has led to a boom of its utilization, including in science and engineering. Chemical research is no longer an exception in this development [1], and numerous areas have been identified, in which ML is now employed to great effect (see, e.g., Refs. [2–7]). While ML applications have resulted in a number of exciting and valuable studies that have advanced chemical domain knowledge, it is worth noting that there is still a considerable lack of quality control, guidance, uniformity, and established protocols for the successful conduct of such studies. Unlike for other application domains of ML or for other techniques employed in chemistry, there are no decades of experience to build on. Guidelines established in other contexts do not necessarily translate to chemical problem settings.

The choices that define chemical ML models, e.g., with respect to featurization (balancing expressiveness and cost), training data sampling (accounting for data volume limitations, biases, imbalances), ML hyperparameter and model selection (balancing complexity and effectiveness), etc., have a dramatic impact on the resulting models’ predictive performance and range of applicability [8]. So far, the community has mostly relied on ad hoc choices that are unlikely to yield the best possible outcomes. The ability to quantify the quality, reliability, and applicability of ML models via metrics is thus an obvious topic of interest. ML approaches that optimize the model design choices do so by minimizing an error metric (e.g., via a fitness function in an evolutionary algorithm [9, 10]). The comparison of different models on the basis of these metrics can also yield design recommendations, illuminate their implications, and thus result in best practices for different problem scenarios within the chemistry domain. Ultimately, they may serve as the foundation for meta-ML facilities and expert recommender systems as part of ML software tools (e.g., Ref. [11–14]).

II. PIECES OF THE METRICS PUZZLE

For ML regression and classification models, there are numerous statistical metrics (also known as loss function metrics) that can be used to characterize their performance. The notion of ‘no-free-lunch’ [15] in computational complexity and optimization theorizes that the performance of any two methods or algorithms is equivalent when averaged across all possible problems. This theorem applies to various aspects of both model selection and validation in ML as well [16]. Loss function metrics are generally based on the comparison of model predictions $y_{i,\text{pred}}$ and an assumed ground truth $y_{i,\text{true}}$ for a number of instances $i$, which leads to prediction errors $e_i$ (Eqn. A1) and relative prediction errors $r_i$ (Eqn. A2), respectively.

Different metrics illuminate different performance aspects of a model. A clear understanding of the specific information a given metric conveys is a prerequisite to fully harnessing it. Blind reliance on a random (e.g., default or commonly reported) metric is a missed opportunity at best and leads to poor outcomes at worst. While particular metrics may be of greater or lesser importance for different application problems, it is generally worth to consider a compilation of metrics. Individual metrics only yield limited insights and no single metric by itself can fully capture the performance of an ML model. But taken together, different metrics complement each other and – like pieces of a puzzle – paint a comprehensive picture of a model’s quality.

The same metrics with respect to the same ground truth need to be compared between different models or studies, otherwise the comparison is meaningless. As an
III. METRICS FOR MODEL VALIDATION AND BENCHMARKING

A. Regression Tasks

For regression tasks, the mean absolute error (MAE) and root mean square error (RMSE) are two of the most commonly reported error metrics (Eqns. A4, A5), and a number of studies have been published debating the supremacy of one over the other [18–24]. (Note that mean absolute deviation (MAD) and root mean square deviation (RMSD) are sometimes used synonymously with MAE and RMSE, respectively. However, since these abbreviations are also used for other statistical metrics such as median absolute deviation or with other definitions, we do not recommend their use to avoid confusion or erroneous conclusions.) The MAE (also called mean unsigned error (MUE)) provides straightforward information about the average magnitude of errors to be expected from a model. However, as all errors are weighted equally, differences in the magnitudes of errors get averaged out, i.e., the MAE alone does not offer insights into the uniformity or variability of prediction errors (and thus the reliability of particular predictions). Metrics that rely on squared errors, such as the RMSE or the less frequently reported mean square error (MSE), magnify larger errors and are thus more sensitive to outliers (which are signaled by large RMSE values). Considered together, MAE and RMSE can yield information on the homogeneity or heterogeneity of errors: if MAE and RMSE values are similar, this indicates prediction errors of relatively consistent magnitude; if the RMSE is significantly larger than the MAE, this indicates large fluctuations in the error magnitudes [25].

MAE and RMSE provide absolute errors that are decoupled from the prediction values. However, the same absolute error has very different implications for smaller or larger prediction values. The mean absolute percentage error (MAPE) and root mean square percentage error (RMSPE) given by Eqns. A6 and A7, respectively, provide error metrics that are relative to the prediction values, and thus complement the absolute MAE and RMSE values. The comparison of MAPE and RMSPE allow us to gauge the uniformity of prediction errors across the range of prediction values (rather than their absolute uniformity; note that absolute and relative uniformity will generally not be achievable at the same time, unless the range of prediction values is very narrow). Use-cases are limited to non-zero prediction values [26–29].

The unsigned errors discussed so far only consider error magnitudes, but not their directional distribution around the prediction. The mean error (ME) and mean percentage error (MPE) given by Eqns. A8 and A9, respectively, allow us to identify systematic biases in the directionality of errors. Unbiased absolute and relative errors have ME and MPE values of 0.0. Positive ME and MPE values indicate systematic overpredictions and negative ones systematic underpredictions. Their magnitude corresponds to the degree of directional bias.

All metrics considered so far provide average errors. They can be complemented by the maximum absolute error (MaxAE) and maximum absolute percentage error (MaxAPE) given by Eqns. A10 and A11, respectively, as well as the difference of most extreme errors ΔMaxE (Eqn. A12), i.e., the spread between largest positive and negative errors. These three metrics provide absolute and relative worst cases in the observed prediction errors. Comparing the maximum error metrics with their corresponding means indicates the degree of deviation between them.

We can further characterize the absolute and/or relative prediction error distributions. Ideally, these should be normal distributions centered around 0.0 with narrow standard deviations σ (Eqn. A13), i.e., the square root of the variance σ². The center of the error distributions are ME and MPE, respectively. A negligible directional bias means that a method is accurate, while small σ means that a method is precise.

We can also quantify the extent of correlation between the prediction results and ground truth by performing a linear regression. The coefficient of determination $R^2$ (Eqn. A14), with $R$ the correlation coefficient, of the fit is a widely reported metric. Maximizing $R^2$ towards the upper limit of 1.0 is equivalent to minimizing the MSE. The slope and offset values of the linear regression (i.e., deviations from 1.0 for the former and 0.0 for the latter) yield additional insights about systematic error behavior that
can complement our findings from the ME, MPE, and σ metrics. Instead of the $R^2$ value, some studies report the adjusted coefficient of determination $R^2_{\text{adj}}$ (Eqn. A15), which incorporates a measure of model complexity, thus giving information about the quality/complexity ratio. While the $R^2$ increases monotonously with the number of features or variables added to a model, the $R^2_{\text{adj}}$ increases only when useful features are added, and decreases otherwise. We could in principle also perform non-linear regressions to further explore the nature of systematic biases, but this is in practice rarely done, as the need for such metrics suggests more fundamental flaws in our ML model. Instead, we could employ ∆-ML or transfer learning techniques to directly correct for the discrepancies between model predictions and ground truth and thus augment and improve the original ML model.

In summary, a good ML model should make predictions with small MAE, RMSE, MAPE, and RMSPE values; small differences between either MAE and RMSE (i.e., homogeneous absolute errors) or MAPE and RMSPE (i.e., homogeneous relative errors); ME and MPE values close to 0.0; small σ; small MaxAE and MaxAPE values with only modest differences to MAE and MAPE, respectively; small ∆MaxE value; $R^2$ and slope close to 1.0 and offset close to 0.0.

### B. Classification Tasks

A simple way of visualizing and reporting the quality of results for classification tasks is via a confusion matrix (Fig. 1) [30], which can be used for both binary and multi-class classifications. A confusion matrix is a square matrix (of size equal to the number of classes) that represents a model’s performance by tabulating class-specific information about the number of correct and incorrect predictions. For a binary classification task, a confusion matrix shows the total number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. These values can be used to calculate other evaluation metrics, including accuracy, precision, and recall.

The simplest of these derived metrics is the accuracy, which is defined as the fraction of correctly labelled predictions among the total number of cases examined (Eqn. A16). While this metric is easy to interpret and suitable for binary and multi-class classification alike, it falls short when dealing with skewed or imbalanced data [31–33]. For cases where the data set is not necessarily balanced, metrics such as precision and recall are preferred [34]. In binary classification problems, precision denotes the fraction of positive class labels that are predicted correctly by the model (Eqn. A17). Recall denotes the overall fraction of the positive class labels that are correctly predicted (Eqn. A18). It is preferred when false negatives are highly undesirable (e.g., if a toxic chemical is falsely predicted to be non-toxic, then it will have far greater ramifications than if a non-toxic chemical is classified as toxic). Thus, in situations where the negative class represents an overwhelming fraction of the training data, precision and recall are more useful than accuracy since it is imperative that all data points belonging to the positive class are predicted correctly. Accuracy, precision, and recall values close to the upper limit of 1.0 are indicative of a well-performing model.

In most cases, there is a trade-off between precision and recall. The $F1$ score, which is the harmonic mean of precision and recall (Eqn. A19), is a useful metric when it is desirable to have a balance between precision and recall [35, 36]. The $F1$ score gives equal weight to precision and recall, however, when domain knowledge or other considerations indicate that more weight should be assigned to one or the other, we can use a weighted $F1$ score ($F_{1,\beta}$), which introduces a weight parameter $\beta$ to adjust the precision-recall trade-off (Eqn. A20).

In certain classification problems, the output of a classifier for a given input is a probability distribution over a set of class labels rather than just the most likely class label. Metrics used to evaluate predicted probabilities are different from those used to evaluate class labels. For predicted probabilities resulting from binary classification, log loss $\mathcal{L}$ (also called binary cross-entropy) (Eqn. A21) is considered a good metric. Although it primarily serves as an objective function for classifiers, it can also be used as an evaluation metric. While it successfully accounts for the uncertainty of a model’s prediction, it needs to be modified with class weights in case of imbalanced data. An extension of this metric for multi-class classifications is the categorical cross-entropy.
While predicted probabilities give a more nuanced view of a classifier’s performance, distinct class labels are preferred for most practical purposes. The latter are derived from the former via a threshold. Two diagnostic metrics (along with domain knowledge) are commonly used to determine the best threshold value, which in turn determines the balance of the classes in the data set. These metrics are the receiver operating characteristic (ROC) curve [37, 38] and the precision-recall (PR) curve [39]. The ROC curve is a plot of the true positive rate (TPR) (Eqn. A18) vs the false positive rate (FPR) (Eqn. A22) at each threshold value. (Note that the TPR is the same as the recall.) The optimum threshold value is one that has a high TPR and a low FPR. Given the ROC curve, we can also compute the area under the ROC curve (AUC) [40–44], which is an important metric used for model selection in classification problems. The closer the AUC value is to the upper limit of 1.0, the better a model performs. We utilize these metrics by plotting the ROC curve with different thresholds and then comparing the AUC for the optimal threshold values for different models.

One shortcoming of ROC curves is that they do not work well for imbalanced or skewed data [45]. For such data sets, PR curves have greater utility [46]. A PR curve is a plot of the model’s precision vs recall at different threshold values. The threshold for which the model has both a high precision as well as a high recall is selected as the optimum value. The F1 score at each threshold can also be determined, along with the area under the PR curve (which is ideally close to 1.0 for a good classifier) and is used for model selection.

In summary, the choice of metrics to assess the quality of an ML classification model depends on the nature of the given data (i.e., balanced or imbalanced), application of the model (which determines the weight to be assigned to positive or negative class labels), and the nature of the classifier itself (i.e., whether it predicts probabilities or individual class labels). As discussed before, it is prudent to compute a set of metrics to obtain a well-developed understanding of a model’s performance.

IV. METRICS FOR UNCERTAINTY QUANTIFICATION

Aside from creating and benchmarking an ML model, an equally important task is to ascertain its applicability to a target domain of interest. For chemistry and drug-related applications, it is common practice to use similarity metrics such as the Tanimoto index T [47] (also called Jaccard coefficient) (Eqn. A23) to gauge the similarity of target molecules to those in the training set. The molecules are numerically represented using fingerprints [48–55]. Similarity in the training and target domains indicates that the predictive performance of the ML model should hold for the target domain.

Formal uncertainty quantification is relatively straightforward if (i) the distribution of the data is known, (ii) the ML model is linear, or (iii) if the model inherently provides an uncertainty for each prediction (such as in Bayesian learning approaches, Gaussian processes, or random forests [56]). If these scenarios do not apply, then we can employ a number of non-parametric, model-agnostic methods to quantify the reliability of predictions made by ML models for a target or ‘query’ point. The perhaps best-known method that has successfully been employed in both regression and classification problems is the ensemble variance (also known as the sensitivity analysis) method. In this method, we create an ensemble of ML models by repeatedly sampling (with replacement) subsets of the training data (also known as bootstrap aggregating or bagging [57]). The variance in their predictions for a query point is used to determine, whether or not the query point lies within the applicability domain [58–61]. The smaller the variance in the predictions, the more likely it is that the query point falls into the applicability domain, whereas larger variances are more likely an indication of the query point being an outlier. Unfortunately, this method has a high computational overhead, in particular with complex models and/or large data, which limits its practical utility.

Another class of methods is based on the range of descriptor values (or those of other representations). For instance, we can examine every descriptor value in the query point with the corresponding range across all points in the training data to assess the applicability of the model to the query point [62]. In geometric methods, we construct convex hulls around the training data to define the extent of the descriptor values. These methods have also been extended to data obtained after a transformation of the initial set of descriptors, such as a representation obtained from a principal component analysis (PCA). However, insights about the density distribution of descriptor values cannot be inferred from range-based methods.

Finally, we can also employ techniques that are distance-based, i.e., they rely on the distance of the query point from the distribution of the training set, assuming that ML predictions are trustworthy in regions of dense data. Distance-based metrics tend to be easy and inexpensive to compute. A model’s applicability domain is determined via a predefined threshold for the distance of a query point from a point within the distribution. This can either be the distance to the mean of the distribution, average (or weighted-average [63, 64]) distance to k-nearest neighbors (neighbors with similar descriptor values) in the training set or the maximum or average distance to all of the points in the distribution. The Euclidean and Mahalanobis distances are the most common distance metrics employed to quantify the distance to a distribution of data points. The Mahalanobis distance indicates the number of standard deviations a query point is away from the mean of a distribution in each dimension that is used to describe the data. These methods have also been adapted to artificial neural networks (in-
V. CONCLUDING REMARKS

In the development and application of ML models, much attention is paid to issues such as the choice of feature representation, data preprocessing, and model selection. While these are all important issues, this review highlights error analysis techniques and metrics as another vital part of ML workflows. The presented analyses and metrics allow us to validate ML models and assess their quality, reliability, and applicability. They also provide the foundation for model development, model comparison, model optimization, and the establishing of guidelines for the deployment of ML in the chemistry domain. Even sophisticated ML models that are trained on very large datasets can easily fail when used without careful consideration of their limitations, and such limitations need to be reported so that potential users are aware of them. The discussed metrics can serve this purpose by illuminating different aspects of the performance of ML models and thus insuring that ML is in a position to advance chemical and materials domain knowledge. The issue of metrics is crucial to further democratize the use of ML in the chemistry community, to promote best practices, to contextualize prediction results and methodological developments, and more broadly to instill the scientific outputs derived from ML work with trust, legitimacy, and transparency.

HIGHLIGHTS

• As machine learning (ML) is gaining an increasingly prominent role in chemical research, so is the need to assess the quality and applicability of ML models, compare different ML models, and develop best-practice guidelines for their design and utilization. Statistical loss function metrics and uncertainty quantification techniques are key issues in this context.

• Different analyses highlight different facets of a model’s performance, and a compilation of metrics - as opposed to a single metric – allows for a well-rounded understanding of what can be expected from a model. They also allow us to identify unexplored regions of chemical space and pursue their survey.

• Metrics can thus make an important contribution to further democratize ML in chemistry, promote best practices, provide context to predictions and methodological developments, lend trust, legitimacy, and transparency to results from ML studies, and ultimately advance chemical domain knowledge.

OUTSTANDING QUESTIONS

• How can we facilitate a greater awareness, appreciation, and education of statistical techniques as well as data science more broadly? It has become clear that there is a need to update traditional curricula in the chemistry domain to account for its rapidly changing research landscape. It has also become clear that these analyses need to be incorporation into ML software tools as prominent features.

• How can we expand our notion of benchmarking and error analysis to put a stronger emphasis on cost-benefit analysis? Given the increasing complexity of ML models that greatly increase their computational demand, it is worth asking if these efforts are actually worthwhile, in particular if they only need to marginal improvements in the predictive performance.

Appendix A: Equations

\[ e_i = y_{i, \text{true}} - y_{i, \text{pred}} \quad (A1) \]

\[ r_i = \frac{y_{i, \text{true}} - y_{i, \text{pred}}}{y_{i, \text{true}}} \quad (A2) \]

\[ \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \quad (A3) \]

\[ MAE = \frac{1}{n} \sum_{i=1}^{n} |e_i| \quad (A4) \]

\[ RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} |e_i|^2} \quad (A5) \]

\[ MAPE = \frac{1}{n} \sum_{i=1}^{n} |r_i| \cdot 100\% \quad (A6) \]

\[ RMSPE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} |r_i|^2} \cdot 100\% \quad (A7) \]
\[ ME = \frac{1}{n} \sum_{i=1}^{n} e_i \]  
(A8)

\[ MPE = \frac{1}{n} \sum_{i=1}^{n} r_i \cdot 100\% \]  
(A9)

\[ \text{MaxAE} = \max \{ e_i \}, \quad i = 1, \ldots, n \]  
(A10)

\[ \text{MaxAPE} = \max \{ |r_i| \cdot 100\% \}, \quad i = 1, \ldots, n \]  
(A11)

\[ \Delta \text{MaxE} = \max \{ e_i \} - \min \{ e_i \}, \quad i = 1, \ldots, n \]  
(A12)

\[ \sigma = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2 \]  
(A13)

\[ R^2 = 1 - \frac{\sum_{i=1}^{n} |e_i|^2}{|y_{i,true} - \bar{y}|^2} \]  
(A14)

\[ R^2_{\text{adj}} = 1 - \frac{(n - 1) \sum_{i=1}^{n} |e_i|^2}{(n - m - 1) \sum_{i=1}^{n} |y_{i,true} - \bar{y}|^2} \]  
(A15)

\[ \text{Acc} = \frac{TP + TN}{TP + FP + FN + TN} \]  
(A16)

\[ \text{Prec} = \frac{TP}{TP + FP} \]  
(A17)

\[ \text{Rec} = \text{TPR} = \frac{TP}{TP + FN} \]  
(A18)

\[ F_1 = 2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} \]  
(A19)

\[ F_{1\beta} = (1 + \beta) \cdot \frac{\text{Prec} \cdot \text{Rec}}{\beta^2 \cdot \text{Prec} + \text{Rec}} \]  
(A20)

\[ \mathcal{L} = -\log P(y_{i,true}|y_{i,pred}) \]
\[ = -(y_{i,true} \log(y_{i,pred}) + (1 - y_{i,true}) \log(1 - y_{i,pred})) \]  
(A21)

\[ FPR = \frac{FP}{FP + TN} \]  
(A22)

\[ T = \frac{w}{u + v - w} \]  
(A23)

with:

\[ \text{Acc} : \text{Accuracy}, \]
\[ \text{Prec} : \text{Precision}, \]
\[ \text{Rec} : \text{Recall}, \]
\[ n : \text{total no. of data points}, \]
\[ m : \text{total no. of features}, \]
\[ u : \text{total no. of features in 1st molecule}, \]
\[ v : \text{total no. of features in 2nd molecule}, \]
\[ w : \text{no. of common features between the 2 molecules} \]

**COMPETING FINANCIAL INTERESTS**

The authors declare to have no competing financial interests.

**ACKNOWLEDGMENTS**

This work was supported by the NSF CAREER program under grant No. OAC-1751161 and the NSF Big Data Spokes program under grant No. IIS-1761990.

**REFERENCES**

[1] Johannes Hachmann, Theresa L. Windus, John A. McLean, Vanessa Allwardt, Alexandra C. Schrimpe-Rutledge, Mohammad Atif Faiz Afzal, and Mojtaba Haghighatlari, Framing the role of big data and modern data science in chemistry, Tech. Rep. (2018) NSF CHE Workshop Report.

[2] Mojtaba Haghighatlari and Johannes Hachmann, “Advances of machine learning in molecular modeling and simulation,” Current Opinion in Chemical Engineering 23, 51–57 (2019).

[3] Mohammad Atif Faiz Afzal, Aditya Sonpal, Mojtaba Haghighatlari, Andrew J. Schultz, and Johannes Hachmann, “A deep neural network model for packing density predictions and its application in the study of 1.5 million organic molecules,” Chemical Science 10, 8374–8383 (2019).

[4] Mohammad Atif Faiz Afzal, Mojtaba Haghighatlari, Sai Prasad Ganesh, Chong Cheng, and Johannes Hachmann, “Accelerated discovery of high-refractive-index polyimides via first-principles molecular modeling, virtual high-throughput screening, and data mining,” The Journal of Physical Chemistry C 123, 14610–14618 (2019).
[5] Mohammad Atif Afzal, Chong Cheng, and Johannes Hachmann, “Combining first-principles and data modeling for the accurate prediction of the refractive index of organic polymers,” The Journal of Chemical Physics 148, 241712 (2018).

[6] Mojtaba Haghighatlarl, Gaurav Vishwakarma, Mohammad Atif Afzal, and Johannes Hachmann, “A Physics-Infused Deep Learning Model for the Prediction of Refractive Indices and Its Use for the Large-Scale Screening of Organic Compound Space,” ChemRxiv, 8796950 (2019).

[7] Mojtaba Haghighatlarl, Ching-Yen Shih, and Johannes Hachmann, “Thinking globally, acting locally: On the issue of training set imbalance and the case for local machine learning models in chemistry,” ChemRxiv, 8796947 (2019).

[8] Mojtaba Haghighatlarl, Making Machine Learning Work in Chemistry: Methodological Innovation, Software Development, and Application Studies, Ph.D. thesis, University at Buffalo (2019).

[9] Gaurav Vishwakarma, Mojtaba Haghighatlarl, and Johannes Hachmann, “Towards autonomous machine learning in chemistry via evolutionary algorithms,” ChemRxiv, 9782387 (2019).

[10] Gaurav Vishwakarma, Machine Learning Model Selection for Predicting Properties of High Refractive Index Polymers, Master’s thesis, University at Buffalo (2018).

[11] Johannes Hachmann, Mohammad Atif Afzal, Mojtaba Haghighatlarl, and Yudhajit Pal, “Building and deploying a cyber-infrastructure for the data-driven design of chemical systems and the exploration of chemical space,” Molecular Simulation 44, 921–929 (2018).

[12] Mojtaba Haghighatlarl, Gaurav Vishwakarma, Doaa Altarawy, Ramachandran Subramanian, Bhargava U Kota, Aditya Sonpal, Sirirangaraj Setlur, and Johannes Hachmann, “ChemML: A machine learning andinformatics program package for the analysis, mining, and modeling of chemical and materials data,” Wiley Interdisciplinary Reviews: Computational Molecular Science 10, e1458 (2020).

[13] Marcus D. Hanwell, Chris Harris, Alessandro Genova, Mojtaba Haghighatlarl, Muammar El Khatib, Patrick Avery, Johannes Hachmann, and Wibe Albert de Jong, “Open chemistry, jupyterlab, rest, and quantum chemistry,” International Journal of Quantum Chemistry, e26472 (2020).

[14] Asela Gunawardana and Guy Shani, “A survey of accuracy evaluation metrics of recommendation tasks,” Journal of Machine Learning Research 10, 2935–2962 (2009).

[15] David H. Wolpert and William G. Macready, “Coevolutionary free lunches,” IEEE Transactions on Evolutionary Computation 9, 721–735 (2005).

[16] Spyros Makridakis, “Accuracy measures: theoretical and practical concerns,” International Journal of Forecasting 9, 527–529 (1993).

[17] Mohammad Atif Faiz Afzal and Johannes Hachmann, “Benchmarking DFT Approaches for the Calculation of Polarizability Inputs for Refractive Index Predictions in Organic Polymers,” Physical Chemistry Chemical Physics 21, 4452–4460 (2019).

[18] Cort J. Willmott and Kenji Matsuura, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance,” Climate Research 30, 79–82 (2005).

[19] Tianfeng Chai and Roland R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature,” Geoscientific Model Development 7, 1247–1250 (2014).

[20] Weijie Wang and Yanmin Lu, “Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model,” in IOP Conference Series: Materials Science and Engineering, Vol. 324 (IOP Publishing, 2018) p. 012049.

[21] Gary Brassington, “Mean absolute error and root mean square error: which is the better metric for assessing model performance?” in EGU General Assembly Conference Abstracts, Vol. 19 (2017) p. 3574.

[22] Cort J. Willmott and Kenji Matsuura, “On the use of dimensioned measures of error to evaluate the performance of spatial interpolators,” International Journal of Geographical Information Science 20, 89–102 (2006).

[23] Cort J. Willmott, “Some comments on the evaluation of model performance,” Bulletin of the American Meteorological Society 63, 1309–1313 (1982).

[24] J. Scott Armstrong and Fred Collopy, “Error measures for generalizing about forecasting methods: Empirical comparisons,” International Journal of Forecasting 8, 69–80 (1992).

[25] Aris A Syntetos and John E Boylan, “The accuracy of intermittent demand estimates,” International Journal of Forecasting 21, 303–314 (2005).

[26] David A. Swanson, Jeff Tayman, and T.M. Bryan, “MAPE-R: a rescaled measure of accuracy for cross-sectional subnational population forecasts,” Journal of Population Research 28, 225–243 (2011).

[27] Louie Ren and Yong Glasure, “Applicability of the revised mean absolute percentage errors (mape) approach to some popular normal and non-normal independent time series,” International Advances in Economic Research 15, 409 (2009).

[28] Jihad Kolassa and Roland Martin, “Percentage errors can ruin your day (and rolling the dice shows how).” Foresee: The International Journal of Applied Forecasting (2011).

[29] Paul Goodwin and Richard Lawton, “On the asymmetry of the symmetric MAPE,” International Journal of Forecasting 15, 405–408 (1999).

[30] Stephen V. Stehman, “Selecting and interpreting measures of thematic classification accuracy,” Remote Sensing of Environment 62, 77–89 (1997).

[31] Nada Lavraˇ c, Peter Flach, and Blaz Zupan, “Rule evaluation metrics of recommendation tasks,” in International Conference on Inductive Logic Programming (Springer, 1999) pp. 174–185.

[32] Qiong Gu, Li Zhu, and Zhihua Cai, “Evaluation measures of the classification performance of imbalanced data sets,” in International Symposium on Intelligence Computation and Applications (Springer, 2009) pp. 461–471.

[33] Mohammad Hessin, Md Nasir Sulaiman, Aida Mustapha, and Norwati Mustapha, “A novel performance metric for building an optimized classifier,” (2011).

[34] Mohammad Furrkranz and Peter A. Flach, “An analysis of rule evaluation metrics,” in Proceedings of the 20th International Conference on Machine Learning (ICML-03) (2003) pp. 202–209.

[35] David Powers, “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation,” Journal of Machine Learning Technologies 2, 37–63
(2011).

[36] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto, *Modern Information Retrieval* (Addison-Wesley Longman Publishing Co., USA, 1999).

[37] Tom Fawcett, “An introduction to ROC analysis,” Pattern Recognition Letters 27, 861–874 (2006).

[38] César Ferri, Peter Flach, and José Hernández-Orallo, “Learning decision trees using the area under the ROC curve,” in *ICML 2002*, Vol. 2 (2002) pp. 139–146.

[39] Takaya Saito and Marc Rehmsmeier, “The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets,” PLOS ONE 10, 1–21 (2015).

[40] David J. Hand and Robert J. Till, “A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems,” Machine Learning 45, 171–186 (2001).

[41] Jin Huang and Charles X Ling, “Using AUC and accuracy in evaluating learning algorithms,” IEEE Transactions on Knowledge and Data Engineering 17, 299–310 (2005).

[42] Alain Rakotomamonjy, “Optimizing Area Under Roc Curve with SVMs.” in *ROCAI* (2004) pp. 71–80.

[43] Peter A. Flach, “The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics,” Proceedings, Twentieth International Conference on Machine Learning 1, 194–201 (2003).

[44] Donna Katzman McClish, “Analyzing a Portion of the ROC Curve,” Medical Decision Making 9, 190–195 (1989).

[45] Chris Drummond and Robert C. Holte, “Cost curves: An improved method for visualizing classifier performance,” Machine Learning 65, 95–130 (2006).

[46] Jesse Davis and Mark Goadrich, “The relationship between Precision-Recall and ROC curves,” in *Proceedings of the 23rd International Conference on Machine learning* (2006) pp. 233–240.

[47] Dávid Bajusz, Anita Rácz, and Károly Héberger, “Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?” Journal of Cheminformatics 7, 20 (2015).

[48] H. L. Morgan, “The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service.” Journal of Chemical Documentation 5, 107–113 (1965).

[49] David Rogers and Mathew Hahn, “Extended-connectivity fingerprints,” Journal of Chemical Information and Modeling 50, 742–754 (2010).

[50] Raymond E. Carhart, Dennis H. Smith, and R. Venkataraman, “Atom pairs as molecular features in structure-activity studies: definition and applications,” Journal of Chemical Information and Computer Sciences 25, 64–73 (1985).

[51] Ramaswamy Nilakantan, Norman Bauman, J. Scott Dixon, and R. Venkataraman, “Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors,” Journal of Chemical Information and Computer Sciences 27, 82–85 (1987).

[52] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nouse, “Reoptimization of MDL Keys for Use in Drug Discovery,” Journal of Chemical Information and Computer Sciences 42, 1273–1280 (2002).

[53] Talete srl, “DRAGON (Software for Molecular Descriptor Calculation),” (2011).

[54] Greg Landrum, “RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling,” (2013).

[55] Noel M. O’Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and Geoffrey R. Hutchison, “Open Babel: An open chemical toolbox,” Journal of Cheminformatics 3, 33 (2011).

[56] Nicolai Meinshausen, “Quantile regression forests,” Journal of Machine Learning Research 7, 983–999 (2006).

[57] Lucas Mentch and Giles Hooker, “Quantifying uncertainty in random forests via confidence intervals and hypothesis tests,” The Journal of Machine Learning Research 17, 841–881 (2016).

[58] Felix Musil, Michael J. Willatt, Mikhail A. Langovoy, and Michele Ceriotti, “Fast and accurate uncertainty estimation in chemical machine learning,” Journal of Chemical Theory and Computation 15, 906–915 (2019).

[59] Andrew A. Peterson, Rune Christensen, and Alireza Khoshidi, “Addressing uncertainty in atomistic machine learning,” Physical Chemistry Chemical Physics 19, 10978–10985 (2017).

[60] Zoran Bosnić and Igor Kononenko, “Comparison of approaches for estimating reliability of individual regression predictions,” Data & Knowledge Engineering 67, 504–516 (2008).

[61] Marko Toplak, Rok Močnik, Matija Polajnar, Zoran Bosnić, Lars Carlsson, Catrin Hasselgren, Janez Demšar, Scott Boyer, Blaž Zupan, and Jonna Staišring, “Assessment of machine learning reliability methods for quantifying the applicability domain of QSAR regression models,” Journal of Chemical Information and Modeling 54, 431–441 (2014).

[62] Joanna Jaworska, Nina Nikulova-Jeliazkova, and Tom Aldenberg, “QSAR applicability domain estimation by projection of the training set in descriptor space: a review,” Alternatives to Laboratory Animals 33, 445–459 (2005).

[63] Rui féng Liu and Anders Wallqvist, “Molecular Similarity-Based Domain Applicability Metric Efficiently Identifies Out-of-Domain Compounds,” Journal of Chemical Information and Modeling 59, 181–189 (2019).

[64] Rui féng Liu and Anders Wallqvist, “Molecular similarity-based domain applicability metric efficiently identifies out-of-domain compounds,” Journal of Chemical Information and Modeling 59, 181–189 (2018).

[65] Jon Paul Janet, Chenrui Duan, Tzuhsiung Yang, Aditya Nandy, and Heather J. Kulik, “A quantitative uncertainty metric controls error in neural network-driven chemical discovery,” Chemical Science 10, 7913–7922 (2019).