CellPress

Trends in Chemistry

Special Issue: Machine Learning for Molecules and Materials

Review

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

Gaurav Vishwakarma,[1,*] Aditya Sonpal,[1] and Johannes Hachmann[1,2,3,*]

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, that is, 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.

Assessing Machine Learning (ML) Models

The rapid advancement and transformation of ML technology has led to a boom in 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., [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 used in chemistry, there are not decades of experience to build upon. Guidelines established in other contexts do not necessarily translate to chemical problem settings.

The choices that define chemical ML models, for example, with respect to featurization (balancing expressiveness and cost), training data sampling (accounting for data volume limitations, biases, imbalances), ML **hyperparameter** (see Glossary) and model selection (balancing complexity and effectiveness), etc., have a dramatic impact on the resulting models' predictive performance and range of applicability. 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** [8]). 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., [9–12]).

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.

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

*Correspondence:
gvishwak@buffalo.edu (G. Vishwakarma) and
hachmann@buffalo.edu (J. Hachmann).

 https://doi.org/10.1016/j.trechm.2020.12.004

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' [13] 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 [14]. Loss function metrics are generally based on the comparison of model predictions $y_{i,\mathrm{pred}}$ and an assumed ground truth $y_{i,\mathrm{true}}$ for a number of instances i, which leads to prediction errors e_i (Equation SI1) and relative prediction errors r_i (Equation SI2), 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 considering 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 alternative to comparing the error metrics of two models (with respect to an independent ground truth), we can also choose the ground truth to be the predictions of one of the models. In that case, the error metrics directly reflect the differences between the two models.

It is important to stress that while error metrics can be applied to the predictions within the training, validation, and test data set (including as part of k-fold cross-validation, in which these sets get reshuffled), only the results for the unseen test set data is considered in the evaluation of the predictive performance of a model. The comparison of training and test set error metrics is instructive as significant differences indicate a poorly trained (e.g., overfitted) model. Similarly, the errors of the different instances of a k-fold cross-validation should be consistent.

In the following sections, we will provide a concise overview of a selection of particularly useful metrics, highlight their advantages and disadvantages, and discuss how a suite of these metrics can afford multifaceted insights into the behavior of a model. It is worth mentioning that much of this discussion is transferable to predictions of non-ML (e.g., physics-based rather than data-derived) models [15]. We also stress that all prediction errors have to be judged in the context of the intrinsic errors or uncertainties of the assumed ground truth.

Metrics for Model Validation and Benchmarking

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 (Equations SI4 and SI5) and a number of studies have been published debating the supremacy of one over the other [16–24]. (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

Glossary

Binary cross-entropy: in a binary classification problem, each sample belongs to either one class or the other (i.e., it has a known probability of 1.0 for one class and 0.0 for the other). A classifier model can estimate the probability of a sample belonging to each class. The binary cross-entropy is used as a metric to assess the difference between the two probability distributions and thus the uncertainty of a classifier's prediction. (Also see cross-entropy, categorical cross-entropy, and log loss.)
Categorical cross-entropy: for multiclass classification problems, that is, for problems involving more than two categories (classes) of data, the cross-entropy measures the difference between the probability distribution of a sample belonging to one class and the probability distribution of that sample not belonging to that class (i.e. belonging to any of the other classes). This metric is known as categorical cross-entropy. (Also see binary cross-entropy.)
Cross-entropy: a measure of the difference between two probability distributions for a given set of samples. (Also see binary cross-entropy, categorical cross-entropy, and log loss.)
Evolutionary/genetic algorithm: This is a heuristic-based approach inspired by natural selection in biological processes (i.e., survival of the fittest). It is typically employed to tackle (combinatorial) optimization problems, in which gradients (needed for gradient descent methods) are ill-defined (e.g., in problems involving discrete or categorical variables) or otherwise inaccessible. Each possible solution behaves as an individual in a population of solutions and a fitness function (itself a loss function metric) is used to determine its quality. Evolutionary optimization of the population takes place via reproduction, mutation, crossover, and selection iterations.
Fitness/objective function: This is a loss function metric that assesses the quality of a solution with respect to an objective of an optimization. Its output can be maximized or minimized (e.g., as part of an evolutionary algorithm).
Harmonic mean: one of multiple types of mean value metrics. Given a set of sample values, the harmonic mean is the inverse of the arithmetic mean of the inverse of the sample values.
Hyperparameter: in ML, hyperparameters are the parameters

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, that is, 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 Equations SI6 and SI7, 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 Equations SI8 and SI9, 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 Equations SI10 and SI11, respectively, as well as the difference of most extreme errors MaxE (Equation SI12), (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 (Equation SI13), i.e., the square root of the variance σ^2. The center of the error and percentage 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 (Equation SI14), 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

that define the structure of a model and control the learning process, as opposed to other parameters that are derived ('learned') from the data in the course of training the model.

Log loss: the negative logarithm of the likelihood of a set of observations given a model's parameters. While log loss and cross-entropy are not the same by definition, they calculate the same quantity when used as fitness functions. In practice, the two terms are thus often used interchangeably.

Loss function metrics: statistical error metrics used to assess the performance of ML models and the quality of their predictions.

Principal component analysis: a technique to transform the feature basis, in which a set of data is described, into a basis that is adapted to the nature of the given data. The principal components are the eigenvectors of the covariance matrix of the data set.

Tanimoto index: this metric is used to assess the similarity between the finite feature (e.g., descriptor, fingerprint) vectors of two samples. The similarity ranges from 0 to 1, with 0 indicating no point of intersection between the two vectors and 1 revealing completely identical vectors.

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_{adj} (Equation SI15), 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_{adj}^2 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 Δ-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.

Figure 1 shows the characteristics and utility of several of the regression metrics discussed in this section for different types of errors.

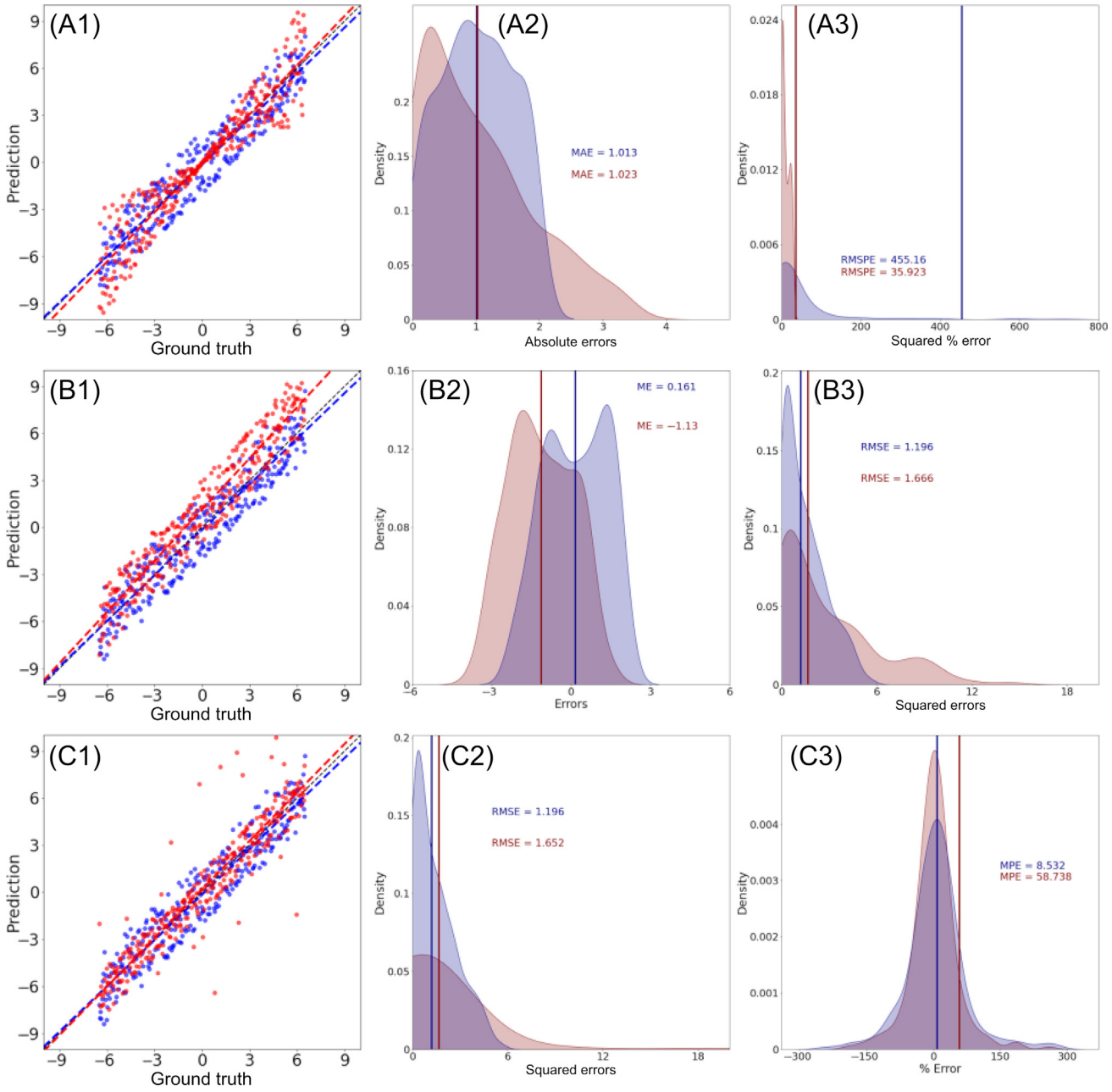
In summary, a good ML model should make predictions with small MAE, RMSE, MAPE, and RMSPE values; small differences between either MAE and RMSE (i.e., homogeneous absolute errors) or MAPE and RMSPE (i.e., homogeneous relative errors); ME and MPE values close to 0.0; small σ; small MaxAE and MaxAPE values 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.

Classification Tasks

A simple way of visualizing and reporting the quality of results for classification tasks is via a confusion matrix (Figure 2) [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 (Equation SI16). 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 (Equation SI17). Recall denotes the overall fraction of the positive class labels that are correctly predicted (Equation SI18). 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 (Equation SI19), 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



Trends in Chemistry

Figure 1. Comparing Pairs of Data Sets with Different Error Characteristics and How These Are Reflected in Different Error Metrics. The data sets each contain 300 data points that are synthetically generated using $y_i = rn\,(kl) \cdot x_i + rn\,(mln)$, with linear and constant random noise rn within the ranges (kl) and (mln), respectively. We use these ranges to simulate different error scenarios. Column 1 shows plots that compare the 'predictions' y, with the ground truth parity line $y = x$ in black. Linear regression lines for each data set are shown in their respective color. Columns 2 and 3 contain error density plots for selected metrics. (A1) Data sets with only linear (red) or only constant errors (blue), chosen such that their mean absolute error (MAE) values are near identical (A2). While the red data has a slightly larger root mean square error (RMSE), the blue data has dramatically larger mean absolute percentage error (MAPE) and root mean square percentage error (RMSPE) (A3). For the red data, MAE and RMSE are similar and MAPE and RMSPE dissimilar, while the situation for the blue data is reversed. (B1) Data sets with a mix of linear and constant error, one where these are directionally biased (red), the other unbiased (blue). The bias in red is reflected in significantly non-zero mean error (ME) (B2) and an incorrect regression slope. As the red data follows the wrong trend, all error metrics are elevated (e.g., RMSE, B3). In (C1), the blue data is the same as in (B1). In the red data,

(Figure legend continued at the bottom of the next page.)

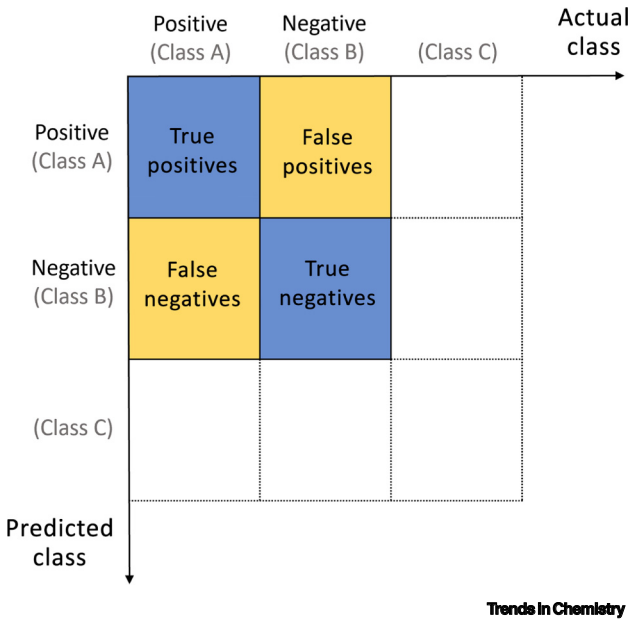


Figure 2. Confusion Matrix for Binary Classification Problems. The matrix can be extended for multiclass classifications where positive and negative classes are replaced with classes A, B, C, etc.

to one or the other, we can use a weighted F1 score ($F1_\beta$), which introduces a weight parameter β to adjust the precision-recall (PR) trade-off (Equation SI20).

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**) (Equation SI21) is considered a good metric. Although it primarily serves as a fitness 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 multiclass classifications is the **categorical cross-entropy** [37,38].

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 [39,40] and the PR curve [41]. The ROC curve is a plot of the true positive rate (TPR) (Equation SI18) versus the false positive rate (FPR) (Equation SI22) 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) [42–46], 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

10% of the data points are replaced with outliers. The error of the remaining data is reduced such that the overall MAE of both sets match. The outliers result in increased RMSE (C2), comparable with that seen for systematic errors (B3). They also by chance led to large non-zero MPE values (C3) that could easily be mistaken for a sign of systematic bias. However, the slope and ME values readily disprove this notion.

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 [47]. For such data sets, PR curves have greater utility [48]. A PR curve is a plot of the model's precision versus 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.

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, where generally the molecules are numerically represented using fingerprints [49–56], it is common practice to use similarity metrics such as the **Tanimoto index** T [57] (also called Jaccard coefficient) (Equation SI23) to gauge the similarity of target molecules to those in the training set. 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) the model inherently provides an uncertainty for each prediction (such as in Bayesian learning approaches, Gaussian processes, or random forests [58]). If these scenarios do not apply, then we can use 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 [59]). The variance in their predictions for a query point is used to determine whether or not the query point lies within the applicability domain [60–63]. 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 [64]. 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, that is, 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 [65,66]) 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 (including deep belief networks) where the distance of the query point from the distribution is measured in the latent space corresponding to the final hidden layer [67]. A quantitative comparison of several methods, including those described earlier, are detailed in [68–71].

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 (see Outstanding Questions). The discussed metrics can serve this purpose by illuminating different aspects of the performance of ML models, 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.

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 incorporated into ML software tools as prominent features, for example, as part of automated recommender systems.

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, which greatly increases their computational demand, it is worth asking if these efforts are actually worthwhile, in particular if they only lead to marginal improvements in the predictive performance.

How can we advance the development and utilization of local rather than global error metrics? Errors are generally not homogeneously distributed across all predictions but will differ in different prediction domains. As not all prediction domains are of equal interest, it is important to further advance methods that can gauge the quality of predictions where we are primarily interested in them. In chemical applications, this is often in extreme (potentially extrapolative) value regions with sparse training data and significantly larger than average errors.

How can we better harness our knowledge of the mathematical structure of different ML models to contextualize the specific error behavior of these models? Can we correlate the nature of latent variables, parametrization, model robustness, etc., with the predictive performance of the corresponding ML model?

Supporting Material

Metrics for Regression Tasks

$$e_i = y_{i,true} - y_{i,pred} \quad [SI1]$$

$$r_i = \frac{y_{i,true} - y_{i,pred}}{y_{i,true}} \quad [SI2]$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \quad [SI3]$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |e_i| \quad [SI4]$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |e_i|^2} \quad [SI5]$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|r_i|\cdot 100\% \quad [SI6]$$

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|r_i|^2}\cdot 100\% \quad [SI7]$$

$$ME = \frac{1}{n}\sum_{i=1}^{n}e_i \quad [SI8]$$

$$MPE = \frac{1}{n}\sum_{i=1}^{n}r_i\cdot 100\% \quad [SI9]$$

$$MaxAE = max\{e_i\}, i = 1, \ldots, n \quad [SI10]$$

$$MaxAPE = max\{|r_i|\cdot 100\%\}, i = 1, \ldots, n \quad [SI11]$$

$$\triangle MaxE = max\{e_i\} - min\{e_i\}, i = 1, \ldots, n \quad [SI12]$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2} \quad [SI13]$$

$$R^2 = 1 - \sum_{i=1}^{n}\frac{|e_i|^2}{|y_{i,true} - \overline{y}|^2} \quad [SI14]$$

$$R^2_{adj} = 1 - \frac{(n-1)}{(n-m-1)}\sum_{i=1}^{n}\frac{|e_i|^2}{|y_{i,true} - \overline{y}|^2} \quad [SI15]$$

Metrics for Classification Tasks

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \quad [SI16]$$

$$Prec = \frac{TP}{TP + FP} \quad [SI17]$$

$$Rec = TPR = \frac{TP}{TP + FN} \quad [SI18]$$

$$F1 = 2\frac{Prec\cdot Rec}{Prec + Rec} \quad [SI19]$$

$$F1_\beta = (1 + \beta)\frac{Prec\cdot Rec}{\beta^2 Prec + Rec} \quad [SI20]$$

$$\mathcal{L} = -logP(y_{i,true}|y_{i,pred}) = -(y_{i,true}\ log(y_{i,pred}) + (1 - y_{i,true})\ log(1 - y_{i,pred})) \quad [SI21]$$

$$FPR = \frac{FP}{FP + TN} \quad [SI22]$$

$$T = \frac{w}{u + v - w} \quad [SI23]$$

where: Acc = accuracy; Prec = precision; Rec = recall; n = total number of data points; m = total number of features; u = total number of features in 1st molecule; v = total number of features in 2nd molecule; and w = number of common features between the two molecules.

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. Hachmann, J. *et al.* (2018) Framing the Role of Big Data and Modern Data Science in Chemistry. In *NSF CHE Workshop Report*
2. Haghighatlari, M. *et al.* (2019) Advances of machine learning in molecular modeling and simulation. *Curr. Opin. Chem. Eng.* 23, 51–57
3. Afzal, M.A.F. *et al.* (2019) A deep neural network model for packing density predictions and its application in the study of 1.5 million organic molecules. *Chem. Sci.* 10, 8374–8383
4. Afzal, M.A.F. *et al.* (2019) Accelerated discovery of high-refractive-index polyimides via first-principles molecular modeling, virtual high-throughput screening, and data mining. *J. Phys. Chem. C* 123, 14610–14618
5. Afzal, M.A.F. *et al.* (2018) Combining first-principles and data modeling for the accurate prediction of the refractive index of organic polymers. *J. Chem. Phys.* 148, 241712
6. Haghighatlari, M. *et al.* (2019) 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* Published online July 8, 2019. http://doi.org/10.26434/chemrxiv.8796950
7. Haghighatlari, M. *et al.* (2019) Thinking globally, acting locally: on the issue of training set imbalance and the case for local machine learning models in chemistry. *ChemRxiv* Published online July 8, 2019. http://doi.org/10.26434/chemrxiv8796947
8. Vishwakarma, G. *et al.* (2019) Towards autonomous machine learning in chemistry via evolutionary algorithms. *ChemRxiv* Published online September 7, 2019. http://doi.org/10.26434/chemrxiv.9782387.v1
9. Hachmann, J. *et al.* (2018) Building and deploying a cyberinfrastructure for the data-driven design of chemical systems and the exploration of chemical space. *Mol. Simul.* 44, 921–929
10. Haghighatlari, M. *et al.* (2020) ChemML: a machine learning and informatics program package for the analysis, mining, and modeling of chemical and materials data. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 10, e1458
11. Hanwell, M.D. *et al.* (2020) Open chemistry, JupyterLab, REST, and quantum chemistry. *Int. J. Quantum Chem.* 121, e26472
12. Gunawardana, A. *et al.* (2009) A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.* 10, 2935–2962
13. Wolpert, D.H. *et al.* (2005) Coevolutionary free lunches. *IEEE Trans. Evol. Comput.* 9, 721–735
14. Makridakis, S. (1993) Accuracy measures: theoretical and practical concerns. *Int. J. Forecast.* 9, 527–529
15. Afzal, M.A.F. *et al.* (2019) Benchmarking DFT approaches for the calculation of polarizability inputs for refractive index predictions in organic polymers. *Phys. Chem. Chem. Phys.* 21, 4452–4460
16. Willmott, C.J. *et al.* (2005) Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* 30, 79–82
17. Chai, T. *et al.* (2014) Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* 7, 1247–1250
18. Wang, W. *et al.* (2018) Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. *IOP Conf. Ser. Mater. Sci. Eng.* 324, 012049
19. Brassington, G. (2017) Mean absolute error and root mean square error: which is the better metric for assessing model performance? In *EGU General Assembly Conference Abstracts* (19), pp. 3574
20. Willmott, C.J. *et al.* (2006) On the use of dimensioned measures of error to evaluate the performance of spatial interpolators. *Int. J. Geogr. Inf. Sci.* 20, 89–102
21. Willmott, C.J. (1982) Some comments on the evaluation of model performance. *Bull. Am. Meteorol. Soc.* 63, 1309–1313
22. Armstrong, J.S. *et al.* (1992) Error measures for generalizing about forecasting methods: empirical comparisons. *Int. J. Forecast.* 8, 69–80
23. Pernot, P. *et al.* (2020) Impact of non-normal error distributions on the benchmarking and ranking of quantum machine learning models. *Mach. Learn: Sci. Tech.* 1, 035011
24. Pernot, P. *et al.* (2020) Probabilistic performance estimators for computational chemistry methods: systematic improvement probability and ranking probability matrix. I. Theory. *J. Chem. Phys.* 152, 164108
25. Syntetos, A.A. *et al.* (2005) The accuracy of intermittent demand estimates. *Int. J. Forecast.* 21, 303–314
26. Swanson, D.A. *et al.* (2011) MAPE-R: a rescaled measure of accuracy for cross-sectional subnational population forecasts. *J. Popul. Res.* 28, 225–243
27. Ren, L. *et al.* (2009) Applicability of the revised mean absolute percentage errors (MAPE) approach to some popular normal and non-normal independent time series. *Int. Adv. Econ. Res.* 15, 409
28. Kolassa, S. *et al.* (2011) Percentage errors can ruin your day (and rolling the dice shows how). *Foresight: Int. J. Appl. Forecast.* 23, 21–29
29. Goodwin, P. *et al.* (1999) On the asymmetry of the symmetric MAPE. *Int. J. Forecast.* 15, 405–408
30. Stehman, S.V. (1997) Selecting and interpreting measures of thematic classification accuracy. *Remote Sens. Environ.* 62, 77–89
31. Lavrač, N. *et al.* (1999) Rule evaluation measures: a unifying view. In *International Conference on Inductive Logic Programming*, pp. 174–185, Springer
32. Gu, Q. *et al.* (2009) Evaluation measures of the classification performance of imbalanced data sets. In *International Symposium on Intelligence Computation and Applications*, pp. 461–471, Springer
33. Hossin, M. *et al.* (2011) A novel performance metric for building an optimized classifier. *J. Comput. Sci.* 7, 582–590
34. Fürnkranz, J. *et al.* (2003) An analysis of rule evaluation metrics. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 202–209, AAAI
35. Powers, D. (2011) Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation. *J. Mach. Learn. Technol.* 2, 37–63
36. Baeza-Yates, R.A. *et al.* (1999) *Modern Information Retrieval*, Addison-Wesley Longman Publishing

37. Ho, Y. *et al.* (2020) The real-world-weight cross-entropy loss function: modeling the costs of mislabeling. *IEEE Access* 8, 4806–4813
38. Gordon-Rodriguez, E. *et al.* (2020) Uses and abuses of the cross-entropy loss: case studies in modern deep learning. *arXiv* 2011.05231
39. Fawcett, T. (2006) An introduction to ROC analysis. *Pattern Recogn. Lett.* 27, 861–874
40. Ferri, C. *et al.* (2002) Learning decision trees using the area under the ROC curve. In *ICML 2002* (Vol. 2), pp. 139–146
41. Saito, T. *et al.* (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One* 10, 1–21
42. Hand, D.J. *et al.* (2001) A simple generalisation of the area under the ROC curve for multiple class classification problems. *Mach. Learn.* 45, 171–186
43. Huang, J. *et al.* (2005) Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* 17, 299–310
44. Rakotomamonjy, A. (2004) Optimizing area under Roc curve with SVMs. In *ROC Analysis in Artificial Intelligence*, pp. 71–80, ROCAI
45. Flach, P.A. (2003) The geometry of ROC space: understanding machine learning metrics through ROC iso-metrics. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (Vol. 1), pp. 194–201, AAAI
46. McClish, D.K. (1989) Analyzing a portion of the ROC curve. *Med. Decis. Mak.* 9, 190–195
47. Drummond, C. *et al.* (2006) Cost curves: an improved method for visualizing classifier performance. *Mach. Learn.* 65, 95–130
48. Davis, J. *et al.* (2006) The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240, Association for Computing Machinery
49. Morgan, H.L. (1965) The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *J. Chem. Doc.* 5, 107–113
50. Rogers, D. *et al.* (2010) Extended-connectivity fingerprints. *J. Chem. Inf. Model.* 50, 742–754
51. Carhart, R.E. *et al.* (1985) Atom pairs as molecular features in structure-activity studies: definition and applications. *J. Chem. Inf. Comput. Sci.* 25, 64–73
52. Nilakantan, R. *et al.* (1987) Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *J. Chem. Inf. Comput. Sci.* 27, 82–85
53. Durant, J.L. *et al.* (2002) Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.* 42, 1273–1280
54. Landrum, G. (2013) *RDKit: A Software Suite for Cheminformatics, Computational Chemistry, and Predictive Modeling*, SouceForge
55. O'Boyle, N.M. *et al.* (2011) Open Babel: an open chemical toolbox. *J. Cheminformatics* 3, 33
56. Mauri, A. *et al.* (2006) DRAGON software: an easy approach to molecular descriptor calculations. *MATCH Commun. Math. Comput. Chem.* 56, 237–248
57. Bajusz, D. *et al.* (2015) Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. Cheminformatics* 7, 20
58. Meinshausen, N. (2006) Quantile regression forests. *J. Mach. Learn. Res.* 7, 983–999
59. Mentch, L. *et al.* (2016) Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *J. Mach. Learn. Res.* 17, 841–881
60. Musil, F. *et al.* (2019) Fast and accurate uncertainty estimation in chemical machine learning. *J. Chem. Theory Comput.* 15, 906–915
61. Peterson, A.A. *et al.* (2017) Addressing uncertainty in atomistic machine learning. *Phys. Chem. Chem. Phys.* 19, 10978–10985
62. Bosnić, Z. *et al.* (2008) Comparison of approaches for estimating reliability of individual regression predictions. *Data Knowl. Eng.* 67, 504–516
63. Toplak, M. *et al.* (2014) Assessment of machine learning reliability methods for quantifying the applicability domain of QSAR regression models. *J. Chem. Inf. Model.* 54, 431–441
64. Jaworska, J. *et al.* (2005) QSAR applicability domain estimation by projection of the training set in descriptor space: a review. *Altern. Lab. Anim* 33, 445–459
65. Liu, R. *et al.* (2019) Molecular similarity-based domain applicability metric efficiently identifies out-of-domain compounds. *J. Chem. Inf. Model.* 59, 181–189
66. Liu, R. *et al.* (2018) Molecular Similarity-Based Domain Applicability Metric Efficiently Identifies out-of-Domain Compounds. *J. Chem. Inf. Model.* 59, 181–189
67. Janet, J.P. *et al.* (2019) A quantitative uncertainty metric controls error in neural network-driven chemical discovery. *Chem. Sci.* 10, 7913–7922
68. Scalia, G. *et al.* (2020) Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *J. Chem. Inf. Model.* 60, 2697–2717
69. Tran, K. *et al.* (2020) Methods for comparing uncertainty quantifications for material property predictions. *Mach. Learn: Sci. Tech.* 1, 025006
70. Hirschfeld, L. *et al.* (2020) Uncertainty quantification using neural networks for molecular property prediction. *J. Chem. Inf. Model.* 60, 3770–3780
71. Rakhimbekova, A. *et al.* (2020) Comprehensive analysis of applicability domains of QSPR models for chemical reactions. *Int. J. Mol. Sci.* 21, 5542