Evaluating Machine Learning Model Performance

![hero](/engineering-education/evaluating-ml-model-performance/hero.jpg)

[Image source]( <https://images.unsplash.com/photo-1578353022439-8cbcd4439e6a?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1050&q=80>)

Machine learning models are expected to provide accurate and trustworthy predictions. To confidently trust their predictions, it is important to assess how machine learning models generalize on test data. We look at how to evaluate model performance.

### Table of Contents

1. Need to evaluate performance

2. Model evaluation techniques

3. Classification model evaluation metrics

4. Regression model evaluation metrics

### Prerequisites

A general understanding of machine learning is required. For an introduction or refresher on some basic machine learning concepts check out this [article](/engineering-education/supervised-learning-algorithms/).

### Useful Terms

\*\*Training set\*\* – this refers to a subset of a dataset used to build predictive models. It includes a set of input examples that will be used to train a model by adjusting the parameters of the set.

\*\*Validation set\*\* – is a subset of a dataset whose purpose is to assess the performance of the model built, during the training phase. It periodically evaluates a model and allows for fine-tuning of the parameters of the model. It is worth noting that not all modeling algorithms need a validation set.

\*\*Test set\*\* – this is also known as unseen data. It is the final evaluation that a model undergoes after the training phase. A test set is a subset of a dataset used to assess the possible future performance of a model. For example, if a model fits to the training set much better than the test set, overfitting is likely present.

\*\*Overfitting\*\*– refers to when a model contains more parameters than can be accounted for by the dataset. Noisy data contributes to overfitting. The generalization of such models is unreliable since the model learns more than it is meant to from the dataset.

### Why Evaluate Performance?

Machine learning has become integral to our daily lives. We interact with a form of machine learning every single day. Since we truly depend on machine learning models for various reasons, it is important to have models that provide accurate and trustworthy predictions for their respective use cases. We must always look to test how a model generalizes on unseen data. For example, in an enterprise setting, these models need to offer real value to the business by producing the highest levels of performance. But how do we evaluate the performance of a model? For classification problems, a very common and obvious answer is to measure its accuracy. However, in this article, we shall discover that model accuracy is not an effective metric to measure model performance. We shall also look at a few metrics for regression problems. By the end of the article, we shall understand that there are a lot of ways to measure the performance of a model.

### Model Evaluation Techniques

The techniques to evaluate the performance of a model can be divided into two; cross-validation and holdout. Both these techniques make use of a test set to assess model performance.

#### Cross Validation

Cross-validation involves the use of a training dataset and an independent dataset. These two sets are a result of the partitioning of the original dataset. The sets are used to evaluate an algorithm. Let’s explore how.

Firstly, the dataset is split into groups of instances that are equal in size. These groups are called folds. The model to be evaluated is trained on all the folds except one. After training, the model is tested on the fold that was excluded. This process is then repeated over and over again, depending on the number of folds. If there are six folds, the process will be repeated six times. The reason for the repetition is that each fold gets to be excluded and act as the test set. Lastly, the average performance across all folds is measured to get an estimation of how effective the algorithm is on a given problem.

A popular cross-validation technique is the k-fold cross-validation. It uses the same steps as described above. The k, which is a user-specified number, stands for the number of folds. The value of k may vary based on the size of the dataset but as an example, let us use a scenario of 4-fold cross-validation.

The model will be trained and tested four times. Let’s say the first-round trains on folds 1,2 and 3. The testing will be on fold 4. Fort the second round, it may train on folds 1,2, and 4 and test on fold 3. For the third, it may train on folds 1,3, and 4 and test on fold 2. The last round will test on folds 2,3 and 4 and test on fold 1.

The interchange between training and test data makes this method very effective. However, compared to the holdout technique, cross-validation takes more time to run and uses more computational resources.

#### Holdout

It is important to get an unbiased estimate of model performance. This is exactly what the holdout technique offers. To get this unbiased estimate, a model is tested on data that is different from the data it was trained on.

This technique divides a dataset into three subsets; training, validation, and test sets. From the terms we defined at the start of the article, we know that the training set helps the model make predictions and that the test set assesses the performance of the model. The validation set also helps to assess the performance of the model by providing an environment to fine-tune the parameters of the model. From this, the best performing model is selected.

The holdout method is ideal for use when dealing with a very large dataset or when working with really limited time. The holdout method is faster than cross-validation, prevents the model from overfitting, and incurs lower computational costs.

### Model Evaluation Metrics

#### Metrics for Classification Problems

Predictions for classification problems yield four types of outcomes; true positives, true negatives, false positives, and false negatives. We shall define them later on. We look at a few metrics for classification problems.

##### Classification Accuracy

The most common evaluation metric for classification problems is accuracy. It is taken as the number of correct predictions against the total number of predictions made (or input samples). However, as much as accuracy is used to evaluate a model, it is not a clear indicator of model performance.

Classification accuracy works best if the samples belonging to each class are equal in number. Consider a scenario with 97% samples of class X and 3% class Y in a training set. A model can very easily achieve 97% training accuracy by predicting each training sample in class X.

Testing the same model on a test set with 55% samples of X and 45% samples of Y, the test accuracy is reduced to 55%. This is why classification accuracy is not a clear indicator of performance. It provides a false sense of attaining high levels of accuracy.

##### Confusion Matrix

The confusion matrix forms the basis for the other types of classification metrics. It is a matrix that fully describes the performance of the model. A confusion matrix gives an in-depth breakdown of the correct and incorrect classifications of each class.

![confusion](/engineering-education/evaluating-ml-model-performance/confusion.png)

Confusion Matrix [Source]( <https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15>)

The four terms represented in the image above are very important. Let’s define them:

\*\*True positives\*\* – a scenario where positive predictions are actually positive.

\*\*True negatives\*\* – negative predictions are actually negative.

\*\*False positives\*\* – positive predictions are actually negative.

\*\*False negatives\*\* – a scenario where negative predictions are actually positive.

The green shapes in the image represent when the model makes the correct prediction. The blue ones represent scenarios where the model made the wrong predictions.

We can calculate accuracy from the confusion matrix. The accuracy is given by taking the average of the values in the “true” diagonal. That is:

Accuracy = (True Positive + True Negative) / Total Sample

Which translates to:

Accuracy = Total Number of Correct Predictions / Total Number of Observations

##### F-Score

F-score is a metric that incorporates both the precision and recall of a test to determine the score. It is the harmonic mean of recall and precision. F-score is also known as F-measure or F1 score. Let’s define precision and recall.

Precision refers to the number of true positives divided by the total positive results predicted by a classifier.

Precision = True Positives / (True Positives + False Positives)

On the other hand, recall is the number of true positives divided by all the samples that should have been predicted as positive.

Recall = True Positives / (True Positives + False Negatives)

![precisionrecall](/engineering-education/evaluating-ml-model-performance/percisionercall.png)

Precision vs Recall [Source]( <https://en.wikipedia.org/wiki/Precision_and_recall>)

In addition to robustness, the F-score shows us how precise a model is by letting us know how many correct classifications are made. The F-score ranges between 0 and 1. The higher the F-score, the greater the performance of the model.

F-score =    2 \* Precision \* Recall

                       Precision + Recall

#### Metrics for Regression Problems

Classification models deal with discrete data. The already covered metrics are ideal for classification tasks since they are concerned with whether a prediction is correct or not. There is no in-between; for example, an output like “fairly correct”. Regression models, on the other hand, deal with continuous data. Predictions are in a continuous range. This is the distinction between the metrics for classification and regression problems. We look at a couple of regression metrics.

##### Mean Absolute Error

The mean absolute error represents the average of the absolute difference between the original and predicted values. Mean absolute error provides the estimate of how far off the actual output the predictions were. However, since it’s an absolute value, it does not indicate the direction of the error.

Mean absolute error is given by:

![mae](/engineering-education/evaluating-ml-model-performance/mae.jpg)

##### Mean Squared Error

The mean squared error is quite similar to the mean absolute error. However, mean squared error uses the average of the square of the difference between original and predicted values. Since this involves the squaring of the errors, larger errors are very notable.

Mean squared error is given by:

![mse](/engineering-education/evaluating-ml-model-performance/mse.jpg)

Mean Squared Error Formula

##### Root Mean Squared Error

The root mean squared error computes the goodness of fit by calculating the square root of the average of squared differences between the predicted and actual values. It is a measure of the average error magnitude.

### Winding Up

In addition to looking at model evaluation techniques, we have gone over a few important metrics for classification and regression problems. An important takeaway is that, for classification problems, classification accuracy is not an effective indicator of model performance. However, it must be noted that some of these metrics depend on the context of the problem that needs to be solved. It is therefore important to understand the problem and the most effective metrics to use to evaluate the model. Good luck!

### References and Further Reading

1. [Introduction to Machine Learning Model Evaluation](<https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f#:~:text=The%20above%20issues%20can%20be%20handled%20by%20evaluating,accuracy%20of%20a%20model%20on%20future%20%28unseen%2Fout-of-sample%29%20data>.)
2. [Metrics to Evaluate your Machine Learning Algorithm]( <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>)
3. [Various ways to evaluate a machine learning model’s performance](<https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15>)
4. [Evaluating a machine learning model.](<https://www.jeremyjordan.me/evaluating-a-machine-learning-model/>)
5. [Machine Learning — How to Evaluate your Model?](<https://towardsdatascience.com/machine-learning-how-to-evaluate-your-model-1dabbdc849a4>)
6. [How to Evaluate Machine Learning Algorithms](<https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/>)