HARVARD | BUSINESS | SCHOOL

9-621-045
REV: SEPTEMBER 21, 2020

MICHAEL W. TOFFEL
NATALIE EPSTEIN
KRIS FERREIRA
YAEL GRUSHKA-COCKAYNE

Assessing Prediction Accuracy of Machine Learning Models

Machine learning refers to a set of algorithms that detect patterns in a dataset in order to make predictions. This note focuses on machine learning models that predict the value of an outcome variable (or dependent variable, often denoted y) based on one or more input variables (or independent, explanatory, or predictor variables, often denoted x_1, x_2, and so on). Such models are developed using data where we can observe both x's and y; this type of machine learning is often referred to as *supervised* machine learning. Each of the outcome and input variables can be numeric variables (price, quantity, throughput time, output rate, number of users, etc.) or categorical variables that take one of a limited number of possible values (yes/no, 0/1, pass/fail, train/bus/airline, etc.). Categorical variables that have just two choices are called binary variables, where values can be coded 1 (positive) or 0 (negative), such as whether an image of a supermarket shelf space depicts a two-liter bottle of Coca Cola (1) or not (0), or whether an X-ray CAT scan image depicts a tumor (1) or not (0).

You may be familiar with several statistical models, such as linear and logistic regression, that are also commonly used as machine learning models, but it is worthwhile to note that the objectives and applications of these models are different in each domain. In a traditional causal analysis using such a model, the objective is to develop unbiased coefficients that *explain* the relationship between each input variable x and the outcome variable y. To do so, the analyst makes deliberate decisions about which control variables, higher order terms (such as x_1^2 or x_1^3), and interactions (such as $x_1 * x_2$) to include in the model. The model then reports this relationship between each x and the y as a series of coefficients (denoted β_1, β_2, and so on).

Machine learning, in contrast, has a different objective: to most accurately *predict* the value of y using the input variables. Rather than an analyst deciding which input variables to include, machine learning automates the selection of which x's to include in the model, which often incorporates more complex relationships in the data; in what follows immediately, we highlight how such decisions are typically made. The model's prediction accuracy is assessed using a series of metrics, the most common of which are described in this note. Different metrics are used to evaluate machine learning models that predict continuous vs. categorical outcome variables, so we discuss each one in a separate section.

HBS Professor Michael W. Toffel, Doctoral Student Natalie Epstein, HBS Professor Kris Ferreira, and Professor Yael Grushka-Cockayne (University of Virginia Darden School of Business) prepared this note as the basis for class discussion.

Avoiding Overfitting

Models are built based on patterns, or relationships, discerned from historical data where we know the values of observations' input and output variables, with the intent to be used to predict observations where we know the values of input variables but not of the outcome variable. As such, we need to balance how closely the model characterizes historical data patterns with how likely it is to predict future ones. The model can go "too far" to try to accurately predict the past observations—such as via a complex model that includes many squared and cubed terms as well as innumerous interaction terms—which can result in the model doing a poor job of predicting new observations, a situation referred to as *overfitting* to the data we have.

A common technique to avoid overfitting is to compare the accuracy of models based on how well they predict observations that were not used to develop the model. To do so, input and output data is often randomly divided into two non-overlapping subsets, commonly referred to as the training dataset and testing dataset (sometimes called validation dataset).

- The *training dataset* is used to train prediction models, which results in parameter values (such as regression coefficients) or rules (such as splitting values in classification and regression tree [CART] models). Model predictions of observations in this data are referred to as "in-sample" because they are within the sample (training dataset) used to construct the model in the first place.

- The *testing dataset* is then used to compare the accuracy of the various trained models. Specifically, the trained models are provided the testing dataset's input variables values to predict the output variables. These predicted outcome values are compared to the actual outcome values from the testing dataset, and various prediction accuracy metrics are applied and compared across different models. High model accuracy (specifically, models with comparatively low prediction error) is a key consideration in choosing the final model, in addition to other important considerations such as model interpretability, model implementation, and maintenance costs.[1]

The final model can then be applied to new observations containing only input variable values in order to predict those observations' outcome values.

Cross Validation

When randomly splitting the dataset into two subsets in order to train (build) and then evaluate our models, concerns may arise—especially when dealing with smaller datasets—that the two subsets might not be sufficiently similar. For example, one of the subsets might, by chance, contain mostly male observations and the other mostly female observations, or one with a majority of people who smoke and the other with very few who smoke. The technique called cross validation helps address this concern.

[1] Dividing the dataset into two subsets is appropriate if your goal is to choose the best of several models being considered. However, if you also want to estimate how accurate your chosen model will be in practice, with new data, then the dataset needs to instead be divided into three subsets, which are typically called the training, validation, and testing datasets. In this scenario, the training dataset is still used to train each model (that is, determine model parameters) being considered. The validation dataset is used to adjust parameters and identify the most accurate model. The chosen model is then often retrained on both the training and validation datasets, and then the testing dataset is used just once to estimate how accurate that final model will be when used in practice.

With cross validation, the overall dataset is divided into several subsets, commonly 5 or 10. Suppose we divide the dataset randomly into 10 subsets. Then the following process is repeated 10 times, using each subset as a testing dataset exactly once: (1) choose one subset to use as the testing dataset, and (2) use the remaining 9 subsets all together as the training dataset. After building the model with the training dataset, calculate and record the accuracy measure (e.g., mean squared error or false positive rate, as described below) on the testing dataset. After implementing this 10 times, the model's accuracy is calculated by averaging the 10 resulting accuracy measures.

So far, we have talked about accuracy but have not explained how accuracy is measured. Below, we describe some common metrics and tools to measure the accuracy of machine learning models that predict continuous outcome variables and models that predict categorical outcome variables.

Assessing the Accuracy of Models that Predict Continuous Outcomes

There are many metrics that can be used to assess the prediction accuracy of models developed using the training dataset and applied to generate predictions of continuous variables in the testing dataset. Three common metrics are *mean squared error* (MSE), *mean absolute deviation* (MAD), and *mean absolute percentage error* (MAPE); equivalent metrics that use the median instead of the mean are also common. These metrics are each based on the difference between the outcome variable's actual value (y_i) and predicted value ($\hat{y}_i$) calculated for each observation *i* in the testing dataset using the model (e.g., regression coefficients) developed from the training dataset.[2]

Mean squared error MSE refers to the average, across all observations in the testing dataset, of the squared difference between each observation's actual value and predicted value. First, for each observation *i* in the testing dataset, calculate the difference between its actual value (y_i) and its predicted value ($\hat{y}_i$), and square this difference:

$$\textit{Squared error}_i = (y_i - \hat{y}_i)^2$$

Then calculate the average (mean) of these squared errors among all observations in the testing dataset. Using *N* to denote the number of observations in the testing dataset, MSE is calculated as:

$$\textit{Mean squared error (MSE)} = \frac{\sum_{i=1}^{N}(Squared\ error_i)}{N} = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}$$

Mean absolute deviation MAD refers to the average, across all observations in the testing dataset, of the absolute difference between each observation's actual value and predicted value. This differs from MSE only because it is based on the absolute value of the difference (often referred to as *distance*) between each testing dataset observation and its predicted value, instead of the square of the distance. First, calculate the absolute deviation for each observation in the testing dataset, which is the absolute value of the difference between the outcome variable (y_i) and the prediction of that outcome variable ($\hat{y}_i$):

$$\textit{Absolute deviation}_i = |\, y_i - \hat{y}_i \,|$$

Then, calculating the average of these differences is straightforward:

[2] The prediction accuracy of regression models, a technique used both in machine learning and to build causal models, is often assessed using *R-squared* (often denoted R^2) or *adjusted* R^2, but those metrics are typically calculated to assess in-sample prediction accuracy. In contrast, out-of-sample prediction accuracy is the focus of the metrics highlighted in this note.

$$\textit{Mean absolute deviation (MAD)} = \frac{\sum_{i=1}^{N}(Absolute\ deviation_i)}{N} = \frac{\sum_{i=1}^{N}(|\ y_i - \hat{y}_i\ |)}{N}$$

Mean absolute percentage error MAPE refers to the average, across all observations in the testing dataset, of the absolute percentage difference between each observation's actual value and predicted value. First, calculate absolute percentage error for each observation in the testing dataset:

$$\textit{Absolute percentage error}_i = \left(\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%\right)$$

Then, calculate the average of these values:

$$\textit{Mean absolute percentage error (MAPE)} = \frac{\sum_{i=1}^{N}(Absolute\ percentage\ error_i)}{N} = \frac{\sum_{i=1}^{N}\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%\right)}{N}$$

In many cases, comparing the prediction accuracy of multiple models using different metrics may lead to different results regarding which model is the most accurate. For example, the model that has the smallest MSE may be different from the model that has the smallest MAD. The choice of which accuracy metric to use might depend on the context. If one is more concerned about large errors, MSE might be a better metric to rely on because it penalizes large errors more than MAD. If relative error matters, such as 10% off the box-office prediction, regardless of whether the movie made $1 million or $100 million in the box office, then MAPE is the better accuracy metric to use. If one wants to ignore the effect of outlier observations or a few, very poor predictions, the median versions of the metrics would be preferred over the mean. Ultimately, some judgment is needed for the selection of the accuracy metric used.

Assessing the Accuracy of Models That Predict Categorical Outcomes

Models that predict categorical outcome variables are often referred to as *classification models*; models with only two categorical outcomes are often referred to as *binary classification models*, whereas models with three or more categorical outcomes are often referred to as *multiclass classification models*. Several related concepts can be used to assess classification model accuracy: (1) metrics to assess probabilistic predictions; (2) the confusion matrix and the concepts of true positives, true negatives, false positives, and false negatives; (3) accuracy metrics including Type I and Type II error rates, precision, recall, and specificity; (4) receiver operating characteristic (ROC) curves and area under the curve (AUC); and (5) precision-recall curves.

Metrics to Assess Probabilistic Predictions

Whereas some classification methods predict in which category an observation is most likely to be (e.g., issuing a binary prediction of whether a student will pass or fail a class), other methods such as logistic regression predict probabilities (ranging 0 to 1) that the observation is a member of each outcome category. For example, suppose we are using a logistic regression model to predict whether a CAT scan image includes a tumor or not, which is a binary outcome. After training the model, we want to test its prediction accuracy using the testing dataset. For each observation in the testing dataset, the model will provide a probabilistic prediction between 0 and 1. This predicted value represents the probability that the patient has a tumor. A predicted value of 0.3, for example, means that the model predicts there is a 30% chance the patient has a tumor.

Brier score The *Brier score* is an accuracy metric that can be easily calculated for classification models that predict probabilities for binary outcome variables (coded 0 or 1). Akin to mean squared error (described earlier), the Brier score is the average, across all *N* observations in the testing dataset, of the squared difference between each observation *i*'s actual value ($y_i = 0$ or 1) and predicted value ($\hat{y}_i$), which is a probability that ranges from 0 to 1). Note that more accurate models will have *lower* Brier scores.

$$\textit{Brier score} = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}$$

Cross-entropy *Cross-entropy* is a common metric that assesses the accuracy of both binary and multiclass classification models. It is calculated as follows: For each observation, consider the predicted probability of only the actual outcome category, and take the log of that predicted probability value. After doing this for every observation, calculate the average across all of the observations, and multiply the result by negative one. More accurate models have lower cross-entropy values. The general formula to calculate cross-entropy for a classification model with *K* outcome categories and *N* observations in the testing dataset is:

$$\textit{Cross-entropy} = -\frac{\sum_{i=1}^{N}(\sum_{k=1}^{K} D_{ik} * \log p_{ik})}{N}$$

Here, D_{ik} is defined to be 1 if observation *i* is actually in category *k* and 0 otherwise, and p_{ik} is the predicted probability that observation *i* is in category *k.*

The rest of the metrics that we will present to assess the accuracy of models that predict categorical outcomes require predicted probabilities to be classified into categorical predictions (such as "Yes, the model predicts the patient has a tumor" versus "No, the model predicts the patient does not have a tumor"). For binary classification models, this is done by selecting a threshold between 0 and 1. If we choose a threshold of 0.5 (a common approach), predicted values that are at least 0.5 will be classified as "Yes" and that are below 0.5 will be classified as "No." A higher threshold will classify more observations as "No." A lower threshold will classify more observations as "Yes."

Some multiclass classification models, which are used when the outcome variable has three or more categories, predict probabilities for each outcome category. Instead of using a threshold, the predicted category is usually considered to be the one with the largest predicted probability. For example, consider a classification model that predicts how a student will get to school (walk, drive, or take a bus) and for a particular student the model predicted walk with probability 0.4, drive with probability 0.5, and take a bus with probability 0.1. To convert these probabilistic predictions to a categorical prediction, we would choose the category with the largest predicted probability: drive. Note that for each student, these predicted probabilities across all the outcomes always sum to 1, since exactly 1 of the outcome categories must actually occur.

The Confusion Matrix

A confusion matrix is a table that illustrates the accuracy of a machine learning algorithm that predicts a categorical variable, where the number of rows and columns of the table both equal the number of outcome categories. Each row corresponds to observations that are *predicted* to be in a category, and each column corresponds to observations that are *actually* in a category. The value in each cell of the table is the number of observations in the testing dataset that are predicted to be in that row's category and are actually in that column's category. Note that the sum of the cells equals the number of observations in the testing dataset.

An example confusion matrix for a binary classification model is shown in **Figure 1.** For a binary variable, the category coded "1" is often thought of as "positive" and the category coded "0" as negative. For illustration, consider a model that seeks to predict the presence of a tumor in a CAT scan image based on a model that codes the presence of a tumor as "1" and its absence as "0." Using this nomenclature, the actual presence of a tumor is denoted as "positive" and its absence is "negative." (This nomenclature is obviously not meant to provide value judgments, as few would view the presence of a tumor as a good thing; instead, it simply answers the question "Is there a tumor?")

Constructing the confusion matrix for a binary classification model involves categorizing predictions into four cells. These cells depict the number of a model's predictions that were accurately predicted to be positive (true positive), accurately predicted to be negative (true negative), incorrectly predicted to be positive (false positive), and incorrectly predicted to be negative (false negative).

Figure 1 Confusion Matrix

	Actually Positive (1)	**Actually Negative (0)**
Predicted to be Positive (1)	# of true positives	# of false positives
Predicted to be Negative (0)	# of false negatives	# of true negatives

Source: Casewriters.

Note: Each of the four cells contains the number of corresponding observations in the testing dataset.

Two terms refer to accurate predictions for binary classification models:

- A *true positive* occurs when an observation that is actually a "1" (a positive) is accurately predicted to be "1" (a positive). For example, an individual with a tumor is accurately predicted to have a tumor.

- A *true negative* occurs when an observation that is *not* actually a "1" is accurately predicted *not* to be "1." For example, an individual who does not have a tumor is accurately predicted not to have a tumor.

Two terms refer to inaccurate predictions for binary classification models:

- A *false positive* occurs when an observation that is actually a "0" is inaccurately predicted to be "1"; in other words, we incorrectly (falsely) predicted positive for an observation that was actually negative. For example, a patient who does not have a tumor is inaccurately predicted to have a tumor. This classification error might lead to an individual being inappropriately subjected to the risks and expense of medical interventions that are actually unnecessary.

- A *false negative* occurs when an observation that *is* actually a "1" is inaccurately predicted *not* to be "1"; in other words, we falsely predicted negative for an observation that was actually positive. For example, a patient who has a tumor is inaccurately predicted not to have a tumor. This classification error might lead to an individual to be inappropriately declined the opportunity for medical intervention, despite the actual need for it.

Data scientists and analysts seek to minimize false positives and false negatives—or for multiclass classification models, any inaccurate predictions—but need to consider that their costs may not be symmetric. In other words, it might be more important to minimize false negatives than to minimize false positives or vice versa. We now have the building blocks to define several metrics that characterize the accuracy of models that predict a categorical outcome variable.

Metrics to Assess Categorical Prediction Accuracy

Given the prevalence of binary classification models, we first introduce metrics to assess prediction accuracy for binary classification models; in the appendix, we show how a few of these metrics can be extended to multiclass classification models.

The following two metrics measure binary classification model accuracy based on how accurately the model predicts observations that are actually positive.

True positive rate (also called sensitivity or recall) This refers to the proportion of all actual positives that are correctly predicted to be positive.

$$\textit{True positive rate} = \textit{Sensitivity} = \textit{Recall} = \frac{\textit{True positives}}{\textit{Actual positives}} = \frac{\textit{True positives}}{\textit{True positives} + \textit{False negatives}}$$

False negative rate (also called Type II error rate) This refers to the proportion of all actual positives that are inaccurately predicted as negative.

$$\textit{False negative rate} = \textit{Type II error rate} = \frac{\textit{False negatives}}{\textit{Actual positives}} = \frac{\textit{False negatives}}{\textit{True positives} + \textit{False negatives}}$$

Note that the false negative rate is simply 100% minus the true positive rate. In other words, the two metrics sum to 100%.

In contrast to the prior two metrics that examine accuracy in terms of actual positives, one important metric measures accuracy in terms of predicted positives.

Precision Precision measures the accuracy of a model's positive predictions; it is calculated as the proportion of observations that were predicted to be positive that were correctly predicted to be positive.

$$\textit{Precision} = \frac{\textit{True positives}}{\textit{Predicted to be positive}} = \frac{\textit{True positives}}{\textit{True positives} + \textit{False positives}}$$

The following two metrics measure binary classification model accuracy based on how accurately they predict observations that are actually negative.

True negative rate (also called specificity) This refers to the proportion of all actual negatives that are correctly predicted to be negative.

$$\textit{True negative rate} = \textit{Specificity} = \frac{\textit{True negatives}}{\textit{Actually negatives}} = \frac{\textit{True negatives}}{\textit{True negatives} + \textit{False positives}}$$

False positive rate (also called Type I error rate) This refers to the proportion of all actual negatives that are inaccurately predicted as positive. In a sense, this is a "false alarm rate" because it reflects how often the model mistakenly predicts negative observations to be positive.

$$\textit{False positive rate} = \textit{Type I error rate} = \frac{\textit{False positives}}{\textit{Actual negatives}} = \frac{\textit{False positives}}{\textit{True negatives} + \textit{False positives}}$$

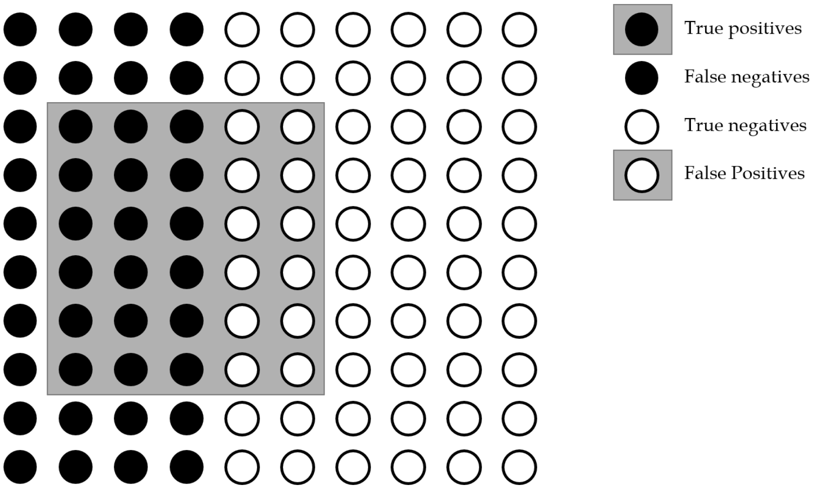
Note that true negative rate is simply 100% minus the false positive rate. In other words, these two metrics sum to 100%.

An illustration Some of the concepts and metrics discussed above are illustrated in **Figure 2**. This depicts an example in which black dots denote the 40 observations that are actually positive, and white dots denote the 60 observations that are actually negative. The gray box distinguishes the model's predictions: dots within the gray box are predicted (or classified) to be positive, and dots outside the gray box are predicted to be negative.

With this setup, we can see that the 18 black dots that are in the gray box represent true positives because these are the actual positive observations that were correctly predicted to be positive. Comparing these to all 40 actual positive observations (all black dots), we can calculate the true positive rate (or sensitivity or recall) as 18/40 = 0.45 = 45%. This means that the false negative rate (or Type II error rate) is 55%, calculated by dividing the 22 false negatives—the black dots that are not covered by the gray box—by the 40 actual positives. Precision is the ratio of the 18 true positives (black dots in the gray box) to all observations predicted to be positives (the 30 dots in the gray box). Thus, precision is 18/30 = 0.6 = 60%.

We can do the same exercise for metrics that evaluate accuracy in terms of predicting actual negatives. The true negative rate (or specificity) refers to the ratio of actual negatives correctly predicted to be negative (true negatives) to all actual negatives. Here, that translates to the ratio of 48 white dots not covered by the gray box (true negatives) to all 60 white dots, or 48/60 = 0.8 = 80%. In contrast, the false positive rate (Type I error rate) is the ratio of actual negatives incorrectly predicted to be positives (that is, the number of false positives) to all actual negatives. In the figure, false positives are depicted as the 12 white dots within the gray box, and so the false positive rate (Type I error rate) is 12/60 = 0.2 = 20%.

Figure 2 Categorical Prediction Accuracy



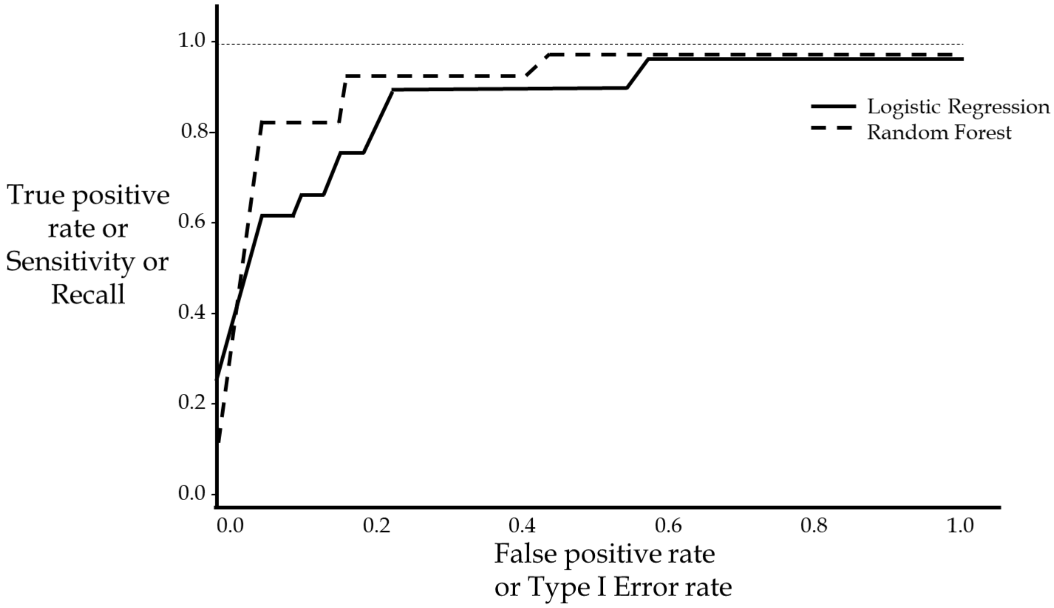
Source: Adapted by casewriters from Michael W. Toffel and Dan Levy, "Improving Worker Safety in the Era of Machine Learning: Introduction to Predictive Analytics," HBS PowerPoint Supplement 619-717 (Boston: Harvard Business School, 2019).

Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC)

As a reminder, for classification methods that predict probabilities as an intermediate step, we must select a threshold to classify predicted probabilities into categorical outcomes. The confusion matrix and associated metrics report values using a single threshold value. A receiver operating characteristic (ROC) curve can be used to evaluate the accuracy of a particular binary classification model across several different thresholds. An ROC curve plots the true positive rate on the y-axis against the false positive rate on the x-axis, for a number of different threshold values (ranging from 0 to 1). A sample pair of ROC curves is depicted in **Figure 3**: one for a logistic regression model (black line) and one for a random forest model (black dashed line). Each point plots the model's accuracy using a specific threshold, and the line connects these points.

Because more accurate models have higher true positive rates and lower false positive rates, we tend to choose a threshold value that produces a point closer to the upper left corner of an ROC curve, as those are the thresholds where the model's predictions are more accurate.

Figure 3 Receiver Operating Characteristic (ROC) Curves for a Logistic Regression Model and a Random Forest Model



Source: Casewriters.

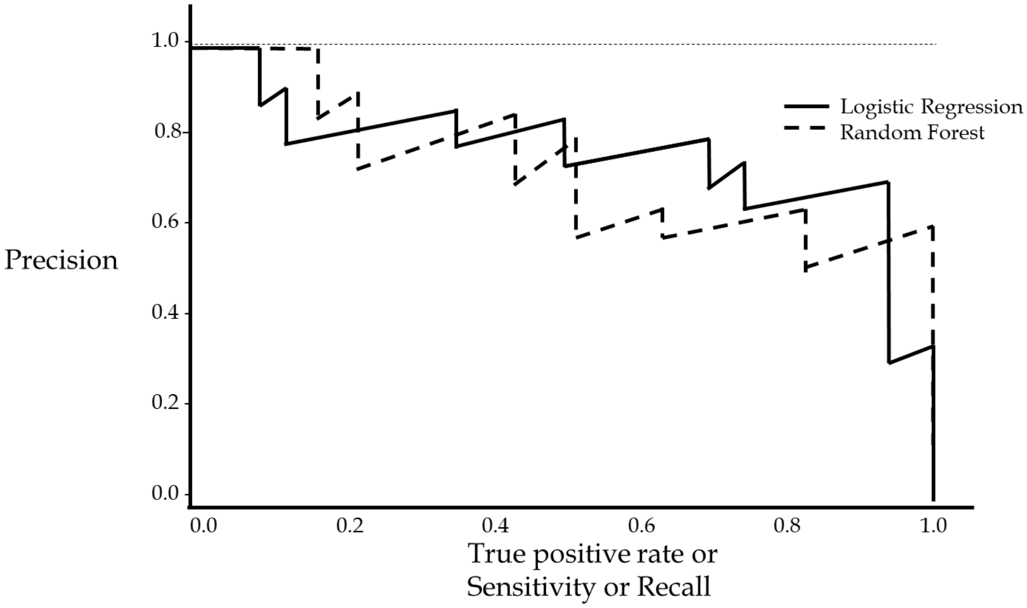
Note: True positive rate (also called recall or sensitivity) is the proportion of all actual positives that are accurately predicted to be positives. False positive rate is the proportion of all actual negatives that are inaccurately predicted as positive. More accurate models bow toward the upper left. The black line is the ROC curve for a logistic regression model, and the black dashed line is the ROC curve for a random forest model.

ROC curves can also be helpful to compare the accuracy of different binary classification models. To do so, we calculate the area under the curve (AUC) score, which is the area under the black line for the logistic regression model or black dashed line for the random forest model in **Figure 3**. ROC curves with higher AUC scores have more points in the upper left of the graph, which means they have more accurate predictions across a range of threshold values; from **Figure 3**, we can see that the random forest model has a greater AUC score than the logistic regression model. A perfectly accurate model would have an AUC score of 1.0. The AUC score is useful to compare models when we have not yet decided what particular threshold to use, perhaps deferring that decision until economic conditions improve or other factors occur. It is also common to compare the overall accuracy of models to ensure one does not choose a model that behaves well only under one specific threshold and very poorly under other plausible thresholds.

Precision-Recall Curve

A precision-recall curve (see **Figure 4**) is another graphical tool used to examine and compare the accuracy of binary classification models across different threshold values.[3] The x-axis is recall, which is the proportion of all actual positives that the model accurately predicted to be positives (also called true positive rate or sensitivity). The y-axis is precision, which is the proportion of all observations the model predicted to be positive that are actually positive. **Figure 4** depicts a sample precision-recall curve for a logistic regression model (black line) and a random forest model (black dashed line). For a specific model, each point plots the model's accuracy using a specific threshold, and the line connects these points. Both high recall and high precision are preferred.

Figure 4 Precision-Recall Curves for a Logistic Regression Model and a Random Forest Model



Source: Casewriters.

Note: Recall (also called true positive rate or sensitivity) is the proportion of all actual positives that are accurately predicted to be positives. Precision is the proportion of all observations predicted to be positive that are actually positive. More accurate models bow toward the upper right. The black line is the precision-recall curve for a logistic regression model, and the black dashed line is the precision-recall curve for a random forest model.

Context matters when deciding on the trade-offs between choosing thresholds and models that yield higher precision or higher recall. Consider the earlier example of a model that predicts whether a CAT scan image contains a tumor. In this context, doctors are more likely to intervene (such as via surgery or chemotherapy) when the model predicts that a CAT scan of a patient reveals a tumor. Models with higher precision are helpful to increase the probability that positive predictions are correct; this minimizes such interventions for patients that do not actually have a tumor (false positives). At the same time, high recall is helpful to avoid missing tumors (predicting no tumor) for patients that actually have them (false negatives).

[3] Although not as commonly used for multiclass classification models, a precision-recall curve could be developed separately for each category.

There is a trade-off in precision versus recall because increasing recall might have doctors intervening for more patients to better ensure they do not miss patients who actually have a tumor. But conducting more medical interventions increases the probability that patients who lack tumors will have a medical intervention, lowering precision, which can be harmful to patients (and wastes resources). This trade-off can be visualized using the precision-recall curve, where thresholds that yield a point closer to the upper right corner of the graph reflect more accurate models.

Further Reading

This note introduced several fundamental metrics and approaches to evaluate the prediction accuracy of machine learning models that use input variables to predict continuous or categorical outcome variables. Those interested in learning more about machine learning may wish to consult these books, both of which are freely available on the web:

- Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. (New York: Springer, 2009, corrected 12th printing January 13, 2017), freely available at https://web.stanford.edu/~hastie/ElemStatLearn/.

- Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R* (New York: Springer, 2013, corrected 7th printing 2017), freely available at http://faculty.marshall.usc.edu/gareth-james/ISL.

Appendix Assessing Accuracy of Multiclass Classification Models

There are three common metrics used to evaluate multiclass classification models: recall, precision, and accuracy. The definitions for recall and precision are analogous to their definitions for binary classification models, where they measured how well the model predicts observations that are actually positive or predicted positive, respectively. Since in multiclass classification there is no notion of positive vs. negative, these metrics are instead calculated for each category. For example, suppose you build a classification model to predict if a student will walk, drive, or take a bus to school. This multiclass classification model has three outcome categories (walk, drive, or take a bus), so $K = 3$.

Recall For each category k, where k refers to any of these outcome categories, *recall* for multiclass models refers to the proportion of all actual observations in category k that are correctly predicted to be in category k. For example, recall for the walk category is the proportion of the students who actually walked that were predicted to walk. More generally, this can be stated as:

$$\textit{Recall for category } k = \frac{\# \, observations \; correctly \; predicted \; in \; category \; k}{total \, \# \, observations \; actually \; in \; category \; k}$$

Precision Similarly, for each category k, where k refers to any of the outcome categories (walk, drive, or take a bus in our example), *precision* for multiclass models refers to the proportion of all observations that were predicted to be in category k that were correctly predicted to be in category k. For example, precision for the walk category is the proportion of the students predicted to walk that actually did walk. It is tempting, but incorrect, to think of precision as the inverse of recall.

$$\textit{Precision for category } k = \frac{\# \, observations \; correctly \; predicted \; in \; category \; k}{total \, \# \, observations \; predicted \; to \; be \; in \; category \; k}$$

To compare recall and precision metrics across different classification models, one approach is to compare each metric individually for each category. For example, you could compare two classification models—for example, a logistic regression model to a random forest model—based on the resulting recall values they produce for the drive outcome category. Another approach is to calculate a model's average recall or precision across all K categories, and then compare different models using these average values.

$$\textit{Average recall for a model} = \frac{\sum_{k=1}^{K} \text{Recall for category } k}{K}$$

$$\textit{Average precision for a model} = \frac{\sum_{k=1}^{K} \text{Precision for category } k}{K}$$

The choice of how to compare these metrics depends on the context. For example, suppose the school system wants to plan bus routes for the upcoming school year. In this case, it might be most important to accurately predict whether the student will take a bus to school, and the data scientist might select the classification model that has the best recall or precision for the "take a bus" outcome category. In contrast, if the school system places equal importance on each of the outcome categories, it might be more appropriate to compare average recall and average precision across various classification models.

Accuracy A third common metric to evaluate multiclass classification models is *accuracy*, which refers to the overall proportion of observations that were correctly predicted.

$$\textit{Accuracy} = \frac{\# \, observations \; correctly \; predicted}{total \, \# \, observations}$$

Although more commonly used to evaluate multiclass classification models, accuracy can also be calculated for binary classification models.