Calculate AUC (Area Under Curve) in R

Authored by

Mohammed looti

November 5, 2025

Evaluating Predictive Power in Binary Classification Models

Logistic Regression remains a cornerstone statistical method across statistics and machine learning, primarily employed for modeling the probability of a dichotomous outcome. When dealing with a response variable that possesses only two states--such as Yes/No or Success/Failure--this model offers a powerful framework for prediction. However, the process of model fitting represents only the initial phase; the subsequent, and arguably more critical, challenge involves rigorously assessing its predictive capabilities.

To accurately evaluate how effectively a logistic regression model generalizes to previously unseen data, data scientists rely on a specialized set of evaluation metrics. A fundamental tool for this assessment is the confusion matrix, from which two of the most insightful metrics are derived: **Sensitivity** and **Specificity**. These metrics provide a nuanced understanding of the model's performance concerning both the positive and negative class predictions.

Sensitivity: Also known as the **True Positive Rate** (TPR), Sensitivity quantifies the proportion of actual positive instances that the model correctly identified. Achieving high sensitivity is crucial in scenarios where minimizing **false negatives** is paramount, such as in the medical diagnosis of a serious condition.

Specificity: Conversely, Specificity, or the **True Negative Rate** (TNR), measures the proportion of actual negative instances that were accurately identified as such. High Specificity is essential when the cost of a **false positive** error is high, for example, incorrectly flagging a low-risk transaction as fraudulent.

While Sensitivity and Specificity offer valuable, concrete insights into performance, they share a critical limitation: their values are inherently dependent upon the specific classification threshold chosen by the analyst. To achieve a comprehensive, threshold-independent measure of a model's overall discriminatory power, the field turns to the **Receiver Operating Characteristic (ROC) curve** and its powerful scalar summary metric, the **Area Under the Curve (AUC)**.

The ROC Curve and the AUC Metric

The ROC curve provides a vital visual diagnostic tool, illustrating the inherent trade-off between **Sensitivity** (True Positive Rate) and **Specificity** (related to the False Positive Rate) across the entire spectrum of potential classification thresholds. This plot places the True Positive Rate on the y-axis against the False Positive Rate (calculated as 1 - Specificity) on the x-axis. An ideal model exhibits a curve that dramatically bends toward the upper-left corner of the plot, signifying high classification accuracy achieved even at very low rates of false positives.

The Area Under the Curve (AUC) distills the complex performance profile summarized by the ROC

curve into a single, easily interpretable value ranging from 0 to 1. Fundamentally, the AUC represents the probability that the trained model will rank a randomly selected positive instance higher than a randomly selected negative instance. This conceptual simplicity makes AUC an extraordinarily robust and powerful metric for evaluating the overall discriminatory ability of any binary classification model, crucially without dependence on a specific cutoff point.

One of the most significant advantages of employing the AUC metric is its independence from issues related to class distribution imbalance. This reliability ensures that the metric provides a fair assessment even when analyzing datasets where one class vastly outnumbers the other. Interpreting the score is intuitive: an AUC value of 1.0 signifies a perfect model that makes no mistakes, while a value of 0.5 suggests the model performs no better than random guessing. Consequently, the closer the AUC score is to 1, the superior the model is at effectively separating the two target classes.

Step 1: Preparing the R Environment and Data Acquisition

To practically demonstrate the calculation of the AUC, we will utilize the powerful R programming language. This demonstration requires a well-suited dataset for binary classification modeling. We have chosen the well-documented **Default** dataset, which is conveniently accessible within the ISLR package. This package is extensively used in educational contexts for teaching fundamental concepts in statistical learning. The Default dataset contains essential information regarding whether various bank customers have defaulted on a loan, alongside crucial predictors such as their student status, credit balance, and annual income.

Before commencing model fitting, standard data science procedure mandates loading the required data and performing an initial structural inspection. The following R commands ensure the dataset is correctly imported and display the first few rows, confirming the readiness of the variables for subsequent analytical steps.

```
#load dataset
data <- ISLR::Default

#view first six rows of dataset
head(data)
```

```
default student balance income
1 No No 729.5265 44361.625
2 No Yes 817.1804 12106.135
3 No No 1073.5492 31767.139
4 No No 529.2506 35704.494
5 No No 785.6559 38463.496
```

6 No Yes 919.5885 7491.559

The resulting output clearly confirms the structure: the dataset includes the critical target variable, `default` (a binary factor: Yes/No), and the three numerical or categorical predictors: `student` status, `balance`, and `income`. These variables form the basis for constructing our predictive logistic regression model.

Step 2: Implementing Data Partitioning and Constructing the Model

Rigorous and unbiased model evaluation necessitates separating the available data into distinct subsets: a **training set** and a **testing set**. The training set is utilized exclusively for estimating the model's coefficients and learning its underlying patterns. Conversely, the testing set is strictly reserved for evaluating the final model's performance on genuinely unseen data. This practice of data partitioning is essential for preventing the common pitfall of **overfitting**, where a model memorizes the training data but fails dramatically when tasked with generalizing to new observations.

In line with standard practice, we allocate 70% of the dataset to the training phase and reserve the remaining 30% for independent testing. To ensure that the random sampling procedure is fully reproducible--a cornerstone of scientific analysis--we invoke the `set.seed(1)` command. This allows any analyst to replicate the exact split and subsequently verify the results.

#make this example reproducible
set.seed(1)

#Use 70% of dataset as training set and remaining 30% as testing set
sample <- sample(c(TRUE, FALSE), nrow(data), replace=TRUE, prob=c(0.7,0.3))
train <- data
test <- data

#fit logistic regression model
model <- glm(default~student+balance+income, family="binomial", data=train)

The core modeling function here is `glm()` (standing for Generalized Linear Model). By specifying the argument `family="binomial"`, we instruct R to fit a logistic regression model, which is appropriate for our binary outcome variable. The formula `default~student+balance+income` defines the predictive relationship, resulting in a trained model object, `model`, now primed to generate probability predictions on the reserved test data.

Step 3: Generating Predictions and Calculating AUC with the pROC Package

After the model has been rigorously trained, the necessary next step involves generating continuous probability scores for every observation within the testing set. The AUC metric requires these probabilities, rather than discrete, hard class predictions (e.g., 'Yes' or 'No'). We achieve this using the standard `predict()` function in R, specifically including the argument `type="response"`, which forces the output to be the estimated probability of the positive class (in this case, the probability of 'Yes' to default).

For the calculation itself, the pROC package stands out as the definitive tool in R for comprehensive analysis and comparison of ROC curves. This package contains the crucial `auc()` function, which substantially streamlines the process. The function requires two primary inputs: the true outcome vector from the test set (the actual responses) and the vector of predicted probability scores generated by the model.

The following concise code block executes the prediction step, loads the required library, and performs the definitive AUC calculation, yielding the final performance metric:

#calculate probability of default for each individual in test dataset
predicted <- predict(model, test, type="response")

#calculate AUC
library(pROC)
auc(test$default, predicted)

Setting levels: control = No, case = Yes
Setting direction: controls < cases
Area under the curve: 0.9437

The resulting output provides the calculated AUC score, which is 0.9437 in this specific demonstration. The messages displayed by the `pROC` package are merely confirmations, indicating that 'No' has been correctly designated as the control (negative class) and 'Yes' as the case (positive class) for the calculation framework.

Interpreting the Model's Performance Based on the Final AUC Score

The final and arguably most important step in model evaluation is contextualizing the magnitude of the calculated AUC score. Since the AUC inherently spans the range from 0 to 1, it offers a direct and easily communicated measure of the model's overall quality and predictive strength.

The result of 0.9437 obtained from our logistic regression model on the Default dataset is

exceptionally high. To provide a standardized framework for interpreting such values, the following general guidelines are often employed within the fields of machine learning and statistical modeling:

0.90 - 1.00: Indicates **Excellent** discriminatory power. The model is highly reliable.

0.80 - 0.90: Indicates **Good** discriminatory power. The model is strong and useful.

0.70 - 0.80: Indicates **Acceptable** discriminatory power. The model provides moderate value.

0.50 - 0.70: Indicates **Poor** discriminatory power. The model is barely better than chance.

0.50 or less: Indicates performance that is equivalent to or worse than random chance.

Given that our model achieved an AUC value of 0.9437, which is strikingly close to 1, we can confidently conclude that the fitted logistic regression performs a tremendously strong job of distinguishing between the two target classes--those individuals who will default on their loan versus those who will not. This high score confirms that the model is highly effective at correctly ranking the likelihood of default for new observations.