

# Learning to Evaluate Logistic Regression Models: A Step-by-Step Guide to Creating ROC Curves in SAS

Authored by  
**Mohammed looti**

October 30, 2025

## RECOMMENDED CITATION

Mohammed looti (2025). *Learning to Evaluate Logistic Regression Models: A Step-by-Step Guide to Creating ROC Curves in SAS*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=6239>

**Logistic regression** stands as a cornerstone statistical technique, particularly indispensable when modeling outcomes where the **response variable** is binary. This means the outcome can only fall into one of two categories--such as "pass/fail," "accepted/rejected," or "yes/no." Unlike its linear counterpart, which forecasts continuous values, logistic regression estimates the probability that a specific event will occur. This fundamental capability makes it a crucial instrument across diverse fields, including medical diagnostics, risk assessment in finance, and behavioral analysis in social sciences, providing clarity on the factors that drive dichotomous results.

When deploying a model designed to categorize outcomes, the critical challenge lies in rigorously assessing its classification accuracy. Simply measuring overall accuracy often proves insufficient, especially when dealing with imbalanced datasets where one outcome is significantly more prevalent than the other. A robust evaluation must therefore delve deeper, examining the model's performance across both positive and negative prediction classes to ensure reliability in real-world applications and avoid misleading conclusions based on simple accuracy rates.

## Evaluating Binary Classification Performance

To gain a comprehensive understanding of a classification model's predictive power, we rely on specialized metrics that quantify its effectiveness in identifying true outcomes. These metrics move beyond simple hit rates, offering granular insights into the model's strengths and potential biases regarding classification errors. Two foundational metrics are essential for this assessment: **Sensitivity** and **Specificity**.

**Sensitivity:** Also known as the **true positive rate** (TPR), sensitivity measures the proportion of actual positive cases that the model correctly flags as positive. In critical applications, such as identifying a rare disease, achieving high sensitivity is paramount because missing a true positive case (a false negative) carries severe consequences. It is the probability of a positive prediction, given that the true outcome is positive.

**Specificity:** Conversely, specificity, or the **true negative rate** (TNR), measures the proportion of actual negative cases that are correctly identified. High specificity is crucial when incorrectly classifying a negative case as positive (a false positive) incurs substantial costs or unnecessary interventions, such as unnecessarily flagging a legitimate transaction as fraudulent. It is the probability of a negative prediction, given that the true outcome is negative.

While these numerical metrics are highly informative, they do not inherently illustrate how performance changes as the classification boundary is adjusted. This necessitates a visual tool that summarizes the inherent trade-off between maximizing true positives and minimizing false positives across all possible thresholds.

## Introduction to the Receiver Operating Characteristic (ROC) Curve

The [Receiver Operating Characteristic \(ROC\) curve](#) provides this essential visual summary. It is a graphical plot illustrating the diagnostic ability of a binary classifier system as its discrimination threshold is systematically varied. The curve is constructed by plotting the true positive rate (Sensitivity) against the false positive rate ( $1 - \text{Specificity}$ ) across every possible probability threshold generated by the model.

The core utility of the [ROC curve](#) is its ability to reveal the crucial trade-off between identifying positive cases (sensitivity) and avoiding false alarms (specificity). Analyzing the curve allows data scientists to select the optimal classification threshold that best balances these two metrics, aligning the model's behavior with the specific risk tolerance and cost structure of the business problem. This tutorial will demonstrate how to generate and interpret this powerful visualization using [SAS](#), one of the leading software suites for statistical analysis.

### Step 1: Preparing the Illustrative Dataset in SAS

The initial stage for any statistical analysis in [SAS](#) involves creating or loading a suitable dataset. For the purpose of demonstrating the [ROC curve](#) generation process, we will define a small, representative dataset containing hypothetical data for 18 students applying to college. This dataset will form the necessary structure for fitting our subsequent [logistic regression](#) model and evaluating its predictive accuracy.

Our dataset, named `my_data`, will incorporate three essential variables relevant to admissions decisions. These variables allow us to model the likelihood of acceptance based on key academic performance indicators:

**Acceptance:** This is our binary [response variable](#). It is coded as **1** for acceptance and **0** for non-acceptance. This categorical structure mandates the use of logistic regression for modeling the probability of the outcome being 1.

**GPA (Grade Point Average):** A continuous [predictor variable](#), typically ranging from 1.0 to 4.0. We anticipate a strong positive correlation between GPA and the probability of acceptance.

**ACT score (American College Testing):** Another quantitative [predictor variable](#) (range 1 to 36). Standardized test scores are critical determinants in the admissions process and are expected to significantly influence the model's predictions.

To construct this dataset directly within the [SAS](#) environment, we employ the `DATA` step combined with the `DATALINES` statement. The `DATA` step initiates the creation of the dataset, and the `INPUT` statement defines the structure of the variables. The `DATALINES` section allows for the direct,

manual entry of the 18 observations, making this approach efficient for small, demonstration-focused examples.

```
/*create dataset*/  
data my_data;  
input acceptance gpa act;  
datalines;  
1 3 30  
0 1 21  
0 2 26  
0 1 24  
1 3 29  
1 3 34  
0 3 31  
1 2 29  
0 1 21  
1 2 21  
0 1 15  
1 3 32  
1 4 31  
1 4 29  
0 1 24  
1 4 29  
1 3 21  
1 4 34  
;  
run;
```

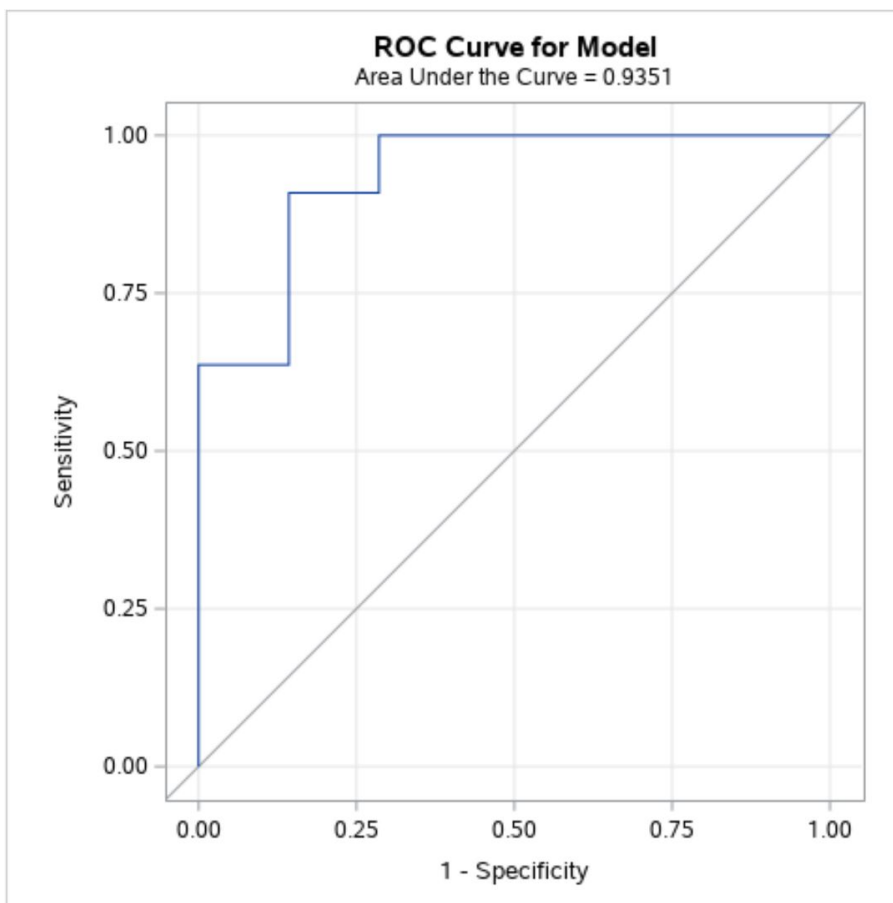
## Step 2: Fitting the Logistic Model and Generating the ROC Plot

Once the data is prepared, we proceed to fit the [logistic regression](#) model using [SAS](#)'s dedicated procedure: [PROC LOGISTIC](#). This powerful procedure is optimized for analyzing categorical response variables and provides comprehensive diagnostics essential for classification tasks.

Within the [PROC LOGISTIC](#) statement, we must specify the model structure. Our `MODEL` statement defines `acceptance` as the outcome variable and `gpa` and `act` as the predictor variables. Crucially, we include the `DESCENDING` option. This instructs [SAS](#) to model the probability of the higher-ordered category (1, representing acceptance). If omitted, the procedure would default to modeling the probability of non-acceptance (0), which is contrary to our primary analytical objective.

To specifically generate the visualization central to this tutorial, the [ROC curve](#), we leverage the `PLOTS` option. Using `PLOTS(ONLY)=ROC` ensures that only the desired plot is produced, offering an immediate visualization of the model's discriminatory capability--that is, how effectively it separates accepted students from non-accepted students across varying probability thresholds.

```
/*fit logistic regression model & create ROC curve*/  
proc logistic data=my_data descending plots(only)=roc;  
model acceptance = gpa act;  
run;
```



### Step 3: Interpreting the ROC Curve and AUC

The visual interpretation of the generated ROC plot is straightforward: a model demonstrating ideal performance will feature a curve that "hugs" the top-left corner of the graph. This geometry indicates that the model is simultaneously achieving high true positive rates (Sensitivity) and low false positive rates ( $1 - \text{Specificity}$ ), signifying exceptional discriminatory power. Conversely, a curve that follows the diagonal line from (0,0) to (1,1) represents a model whose predictions are no better than random chance. The greater the distance between the curve and this diagonal, the

stronger the predictive capability of the model.

While visual assessment offers qualitative assurance, quantitative evaluation is necessary for precise model comparison. This is provided by the [Area Under the Curve \(AUC\)](#). The [AUC](#) is a single scalar metric that summarizes the classifier's overall performance across all possible classification thresholds. Essentially, the [AUC](#) represents the probability that the model will rank a randomly selected positive instance higher than a randomly selected negative instance.

The [AUC](#) score ranges from 0 to 1. A score of **1.0** denotes a perfect classifier capable of separating all positive and negative cases flawlessly. A score of **0.5** means the model is as good as a random guess. Critically, the output generated by [PROC LOGISTIC](#) includes this value directly beneath the plotted curve. For our college acceptance model, the calculated [AUC](#) is **0.9351**. This high value confirms the visual impression of strong performance, indicating that our model is highly effective at distinguishing between successful and unsuccessful applicants based on GPA and ACT scores.

The [AUC](#) metric is particularly valuable when comparing competing classification models. If two distinct [logistic regression](#) models are developed--perhaps leveraging different variables or estimation techniques--the one with the higher [AUC](#) is objectively superior in its overall ability to discriminate between the classes. Consider the following comparison:

AUC of Model A: **0.9351**

AUC of Model B: **0.8140**

In this scenario, Model A, with its significantly higher AUC, is the preferred choice, as it offers demonstrably better discriminatory performance across all possible thresholds, making its predictions more reliable for the college acceptance application.

## Conclusion and Further SAS Resources

Generating the ROC curve and calculating the AUC in [PROC LOGISTIC](#) is a vital step in validating any binary classification model. By understanding the visual representation and quantitative summary provided by the AUC, analysts can confidently select and deploy models that effectively meet business or research objectives.

To further advance your skills in statistical modeling and data manipulation within the [SAS](#) environment, we recommend exploring additional documentation regarding common procedures and specialized analyses: