

# Evaluation Curves for Binary Classification

The two most popular evaluation curves for binary classification are the Receiver Operating Characteristic (ROC) curve and the Precision Recall (PR) curve. For the purposes of this note, we will use the following classification rule:

$$\hat{y} = \begin{cases} 1 & \text{if predicted probability} > \text{threshold} \\ 0 & \text{if predicted probability} \leq \text{threshold} \end{cases}$$

Coordinates for the ROC and PR curves are formed by varying the classification threshold from the maximum possible predicted value to a value just below the minimum possible predicted value.

For the ROC curve, each plot point consists of the following coordinates:

- $x$  = false positive rate
- $y$  = true positive rate

For the PR curve, each plot point consists of the following coordinates:

- $x$  = true positive rate (also known as the recall)
- $y$  = precision

Given a classification threshold, we can use a confusion matrix to compute the false positive rate, true positive rate, and precision values for that threshold as follows:

		Prediction	
		$\hat{y} = 1$	$\hat{y} = 0$
Actual	$y = 1$	True Positive (TP)	False Negative (FN)
	$y = 0$	False Positive (FP)	True Negative (TN)

$$\begin{aligned} \text{False Positive Rate} &= \text{FP} / (\text{FP} + \text{TN}) &= \text{Probability}(\hat{y} = 1 \mid y = 0) \\ \text{True Positive Rate} &= \text{TP} / (\text{TP} + \text{FN}) &= \text{Probability}(\hat{y} = 1 \mid y = 1) \\ \text{Precision} &= \text{TP} / (\text{TP} + \text{FP}) &= \text{Probability}(y = 1 \mid \hat{y} = 1) \end{aligned}$$

It's possible to compare two classifiers based on the Area Under the Curve (AUC), where larger is better. For an ROC curve, the AUC can be interpreted as the probability that a randomly selected positive class observation will have a larger "predicted probability of positive class membership" than a randomly selected negative class observation. For a perfect classifier, we expect the Area Under the Curve (AUC) to be 1 for the ROC curve and approximately 1 for the PR curve (because the minimum true positive rate for the perfect PR curve will be slightly larger than zero). For a classifier that randomly assigns predicted probabilities to a test observation:

- we expect the False Positive Rate to equal the True Positive Rate, so the area under the ROC curve for a random classifier will be 0.5
- we expect the Precision to equal the Base Rate, so the area under the PR curve will be approximately equal to the Base Rate

$$\text{Base Rate} = (\text{TP} + \text{FN}) / (\text{TP} + \text{FN} + \text{FP} + \text{TN}) = \text{Probability}(y = 1)$$

The PR curve is often preferred over the ROC curve when the Base Rate is small.

In practice, we are often tasked with constrained optimization. For example, we may want to maximize the true positive rate subject to the constraint that the false positive rate cannot exceed some maximum value.

```

# an example using R
curves = function(actuals, predictions) {
  thresholds = c(unique(sort(predictions, decreasing = T)), min(predictions) - .Machine$double.eps)
  k = length(thresholds)
  tp = array(0, k)
  fn = array(0, k)
  fp = array(0, k)
  tn = array(0, k)
  for (i in 1:k) {
    tp[i] = sum((actuals == 1) & (predictions > thresholds[i]))
    fn[i] = sum((actuals == 1) & (predictions <= thresholds[i]))
    fp[i] = sum((actuals == 0) & (predictions > thresholds[i]))
    tn[i] = sum((actuals == 0) & (predictions <= thresholds[i]))
  }
  return(data.frame(thresholds, tp, fn, fp, tn,
    fp.rate = fp / (fp + tn),
    tp.rate = tp / (tp + fn),
    precision = tp / (tp + fp)))
}

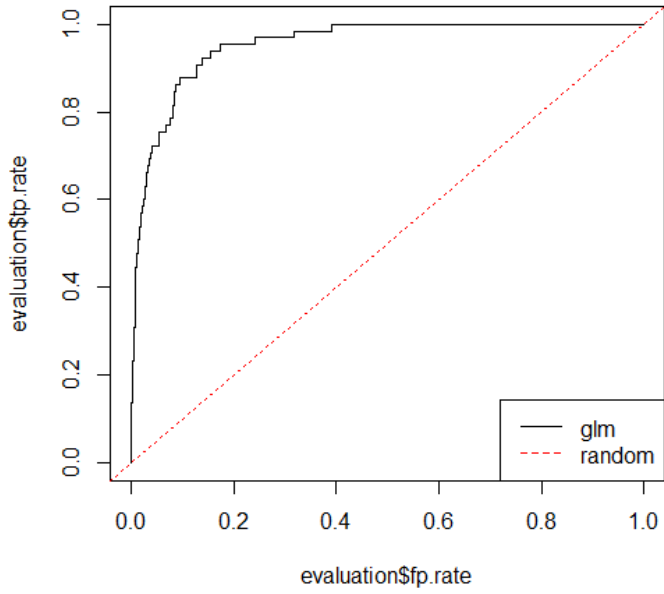
auc = function(x, y) {
  area = 0.0
  for (i in 2:length(x)) {
    # adding the area of the lower rectangle
    # plus half the area of the upper rectangle
    area = area + (x[i] - x[i - 1]) * (min(y[i], y[i - 1]) + 0.5 * abs(y[i] - y[i - 1]))
  }
  return(area)
}

set.seed(2^17-1)
library(ISLR)
index = sample(1:nrow(Default))
trn = Default[index[1:8000],]
tst = Default[index[8001:10000],]
model = glm(default ~ student + balance, data = trn, family = binomial)
predictions = predict(model, newdata = tst, type = "response")

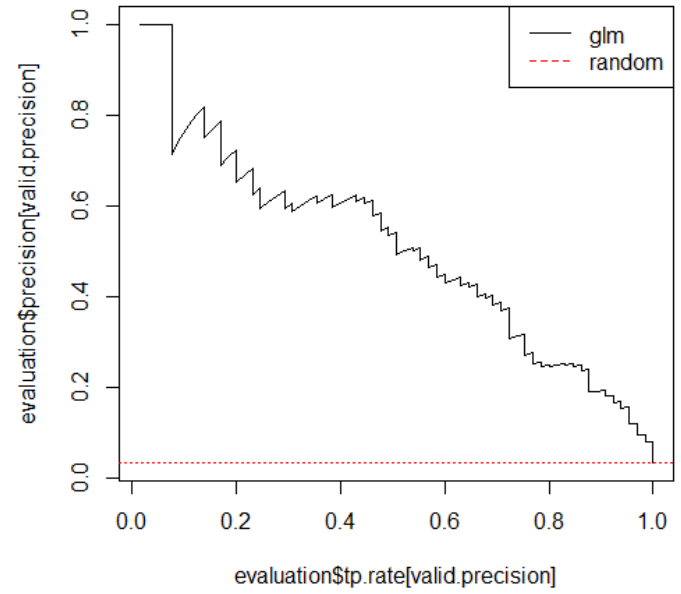
evaluation = curves(as.integer(tst$default == "Yes"), predictions)
valid.precision = 2:nrow(evaluation)
par(mfrow = c(1, 2))
plot(evaluation$fp.rate, evaluation$tp.rate, type = "l")
title(paste("auc =", round(auc(evaluation$fp.rate, evaluation$tp.rate), 3)))
abline(a = 0, b = 1, lty = "dotted", col = "red")
legend("bottomright", legend = c("glm", "random"), lty = c("solid", "dashed"), col = c("black", "red"))
plot(evaluation$tp.rate[valid.precision], evaluation$precision[valid.precision], type = "l")
title(paste("auc =", round(auc(evaluation$tp.rate[valid.precision], evaluation$precision[valid.precision]), 3)))
abline(a = mean(tst$default == "Yes"), b = 0, lty = "dotted", col = "red")
legend("topright", legend = c("glm", "random"), lty = c("solid", "dashed"), col = c("black", "red"))

```

auc = 0.954



auc = 0.495



As shown, the ROC curve and the PR curve are neither strictly concave (line segments joining any two points lie below the curve) nor strictly convex (line segments joining any two points lie above the curve).