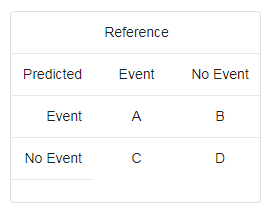
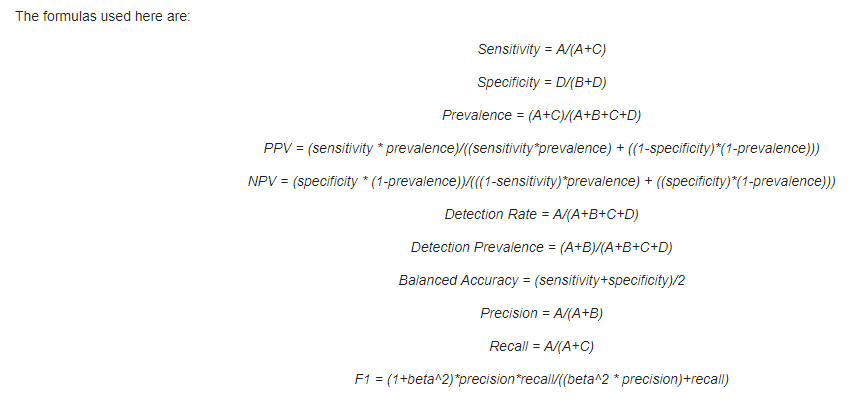
**Details**

The functions requires that the factors have exactly the same levels.

For two class problems, the sensitivity, specificity, positive predictive value and negative predictive value is calculated using the positive argument. Also, the prevalence of the "event" is computed from the data (unless passed in as an argument), the detection rate (the rate of true events also predicted to be events) and the detection prevalence (the prevalence of predicted events).

Suppose a 2x2 table with notation





where beta = 1 for this function.

See the references for discussions of the first five formulas.

For more than two classes, these results are calculated comparing each factor level to the remaining levels (i.e. a "one versus all" approach).

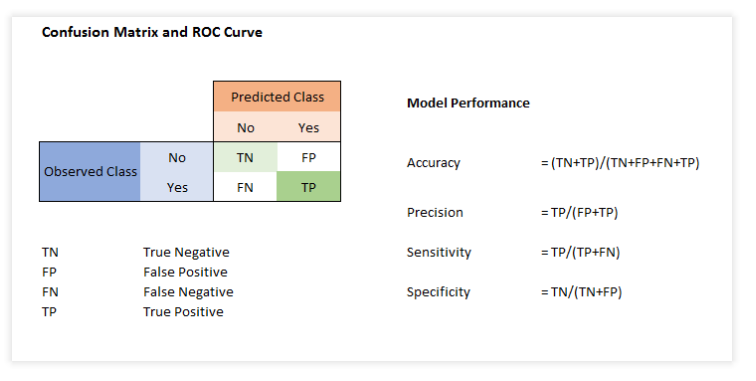
The overall accuracy and unweighted Kappa statistic are calculated. A p-value from McNemar's test is also computed using mcnemar.test (which can produce NAvalues with sparse tables).

The overall accuracy rate is computed along with a 95 percent confidence interval for this rate (using binom.test) and a one-sided test to see if the accuracy is better than the "no information rate," which is taken to be the largest class percentage in the data.

# [Confusion Matrix](http://scaryscientist.blogspot.com/2016/03/confusion-matrix.html)

A confusion matrix is used with classification models, meaning the response value is categorical. Thus, the model predicts one response value for each observation in the testing set, and each predicted response value is compared to the actual response value for that observation. So, confusion matrix assess accuracy of predictive model built.

The confusion matrix is only useful for evaluating a model when we know the true response values. However, once you have put it into production, it is impossible to know which observations are TP/TN/FP/FN unless you have a system for checking the actual response values. The point of the confusion matrix is to give you the expected values for sensitivity/specificity/etc, but there is no guarantee that those rates will come true in production.



**Accuracy** measures overall accuracy of the model classification.

Accuracy = (TP+TN)/(TP+FP+TN+FN)

**Precision** measure accuracy of a class, when predicts "Yes" how often it is correct.

What proportion of positive identifications was actually correct

 For example in a Credit Scoring, class 1 (Good) are given credit facility, Precision measure efficacy of class 1 assignment using the predictive mode. % of the predicted 1's which are actually 1's (based on observed class)

Precision = TP/(TP+FP)  = TP/ (Predicted Yes)

**Sensitivity or Recall**: When it is actually "yes" how often it predicts "Yes".

From the observed class 1, what % cases are actually classified by the model predicted class 1 is measured by a measure which is called Sensitivity.

Recall = Sensitivity = TP/(TP+FN) = TP/ ( Actual Yes)

**Specificity**: Specificity measures true negative rate. When it is actually "No", how often it is "No".

Specificity= TN/(TN+FP) = TN/(Actual No)

**Error rate or Mis classification Rate:** Overall, how often is it wrong?

Error Rate= (FP+FN)/(TP+FP+TN+FN) = 1- Accuracy

**Prevalence:** How often does the yes condition actually occur in our sample?

Prevalence= Actual yes/Total = ( FN+TP)/(TP+FP+TN+FN)

**Positive Predictive Value:** This is very similar to precision, except that it takes prevalence into account. In the case where the classes are perfectly balanced (meaning the prevalence is 50%), the positive predictive value (PPV) is equivalent to precision.

**Null Error Rate:** This is how often you would be wrong if you always predicted the majority class. (In our example, the null error rate would be 60/165=0.36 because if you always predicted yes, you would only be wrong for the 60 "no" cases.) This can be a useful baseline metric to compare your classifier against. However, the best classifier for a particular application will sometimes have a higher error rate than the null error rate.

**Cohen's Kappa:** This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate.

Which is a better metric for evaluating the correctness of the model?

If I have a spam filter (in which the positive class is "spam" and the negative class is "not spam"), I might optimize for precision or specificity because I want to minimize false positives (cases in which non-spam is sent to the spam box). If I have a metal detector (in which the positive class is "has metal"), I might optimize for sensitivity (also known as True Positive Rate) because I want to minimize false negatives (cases in which someone has metal and the detector doesn't detect it).

**caret** package in R has function **confusionMatrix** which is used for calculating a cross-tabulation of observed and predicted classes.

**confusionMatrix** has two important parameters. First one *data* for giving predicted class variable and second *reference* used for giving observed class variable. Both of these variables have to factor type.

*positive* can be used for defining target level to be predicted.

Output of confusionMatrix has a number of different statistics. In the example, 83% is prediction accuracy.

#Install Packages

install.packages("caret"  )

install.packages("e1071")

# Load Library or packages

library(e1071)

library(caret)

# Create Confusion Matrix

confusionMatrix(data=factor(termCrossSell$predicted\_target),

                reference=factor(termCrossSell$target),

                positive='1')

Confusion Matrix and Statistics

          Reference

Prediction     0     1

         0    33225  3607

         1     3323  1033

               Accuracy : 0.8317

                 95% CI : (0.8281, 0.8353)

    No Information Rate : 0.8873

    P-Value [Acc > NIR] : 1.000000

                  Kappa : 0.1353

 Mcnemar's Test P-Value : 0.000675

            Sensitivity : 0.22263

            Specificity : 0.90908

         Pos Pred Value : 0.23714

         Neg Pred Value : 0.90207

             Prevalence : 0.11265

         Detection Rate : 0.02508

   Detection Prevalence : 0.10576

      Balanced Accuracy : 0.56585

       'Positive' Class : 1