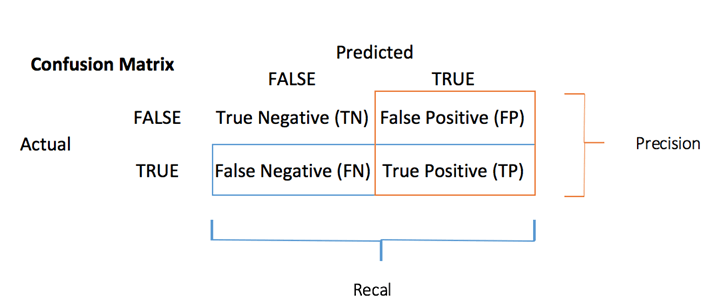
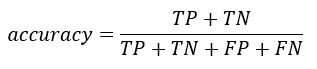
**Four outcomes of the confusion matrix**

The confusion matrix visualizes the accuracy of a classifier by comparing the actual and predicted classes. The binary confusion matrix is composed of squares:

[](https://www.guru99.com/images/r_programming/032918_0938_DecisionTre2.png)Confusion Table

* TP: True Positive: Predicted values correctly predicted as actual positive
* FP: Predicted values incorrectly predicted an actual positive. i.e., Negative values predicted as positive
* FN: False Negative: Positive values predicted as negative
* TN: True Negative: Predicted values correctly predicted as an actual negative

You can compute the **accuracy test** from the confusion matrix:

[](https://www.guru99.com/images/r_programming/032918_0938_DecisionTre3.jpg)

**Example of Confusion Matrix:**

Confusion Matrix is a useful machine learning method which allows you to measure Recall, Precision, Accuracy, and AUC-ROC curve. Below given is an example to know the terms True Positive, True Negative, False Negative, and True Negative.

**True Positive:**

You projected positive and its turn out to be true. For example, you had predicted that France would win the world cup, and it won.

**True Negative:**

When you predicted negative, and it's true. You had predicted that England would not win and it lost.

**False Positive:**

Your prediction is positive, and it is false.

You had predicted that England would win, but it lost.

**False Negative:**

Your prediction is negative, and result it is also false.

You had predicted that France would not win, but it won.

You should remember that we describe predicted values as either True or False or Positive and Negative.

**How to Calculate a Confusion Matrix**

Here, is step by step process for calculating a confusion Matrix in data mining

* Step 1) First, you need to test dataset with its expected outcome values.
* Step 2) Predict all the rows in the test dataset.
* Step 3) Calculate the expected predictions and outcomes:

1. The total of correct predictions of each class.
2. The total of incorrect predictions of each class.

After that, these numbers are organized in the below-given methods:

* Every row of the matrix links to a predicted class.
* Every column of the matrix corresponds with an actual class.
* The total counts of correct and incorrect classification are entered into the table.
* The sum of correct predictions for a class go into the predicted column and expected row for that class value.
* The sum of incorrect predictions for a class goes into the expected row for that class value and the predicted column for that specific class value.

**Other Important Terms using a Confusion matrix**

* **Positive Predictive Value(PVV):** This is very much near to precision. One significant difference between the two-term is that PVV considers prevalence. In the situation where the classes are perfectly balanced, the positive predictive value is the same as precision.
* **Null Error Rate:** This term is used to define how many times your prediction would be wrong if you can predict the majority class. You can consider it as a baseline metric to compare your classifier.
* **F Score:** F1 score is a weighted average score of the true positive (recall) and precision.
* **Roc Curve:** Roc curve shows the true positive rates against the false positive rate at various cut points. It also demonstrates a trade-off between sensitivity (recall and specificity or the true negative rate).
* **Precision:** The precision metric shows the accuracy of the positive class. It measures how likely the prediction of the positive class is correct.

[https://www.guru99.com/images/tensorflow/082918_1030_LinearClass9.jpg](https://www.guru99.com/images/tensorflow/082918_1030_LinearClass9.jpg)

The maximum score is 1 when the classifier perfectly classifies all the positive values. Precision alone is not very helpful because it ignores the negative class. The metric is usually paired with Recall metric. Recall is also called sensitivity or true positive rate.

* **Sensitivity**: Sensitivity computes the ratio of positive classes correctly detected. This metric gives how good the model is to recognize a positive class.

[https://www.guru99.com/images/tensorflow/082918_1030_LinearClass10.jpg](https://www.guru99.com/images/tensorflow/082918_1030_LinearClass10.jpg)

**Why you need Confusion matrix?**

Here are pros/benefits of using a confusion matrix.

* It shows how any classification model is confused when it makes predictions.
* Confusion matrix not only gives you insight into the errors being made by your classifier but also types of errors that are being made.
* This breakdown helps you to overcomes the limitation of using classification accuracy alone.
* Every column of the confusion matrix represents the instances of that predicted class.
* Each row of the confusion matrix represents the instances of the actual class.
* It provides insight not only the errors which are made by a classifier but also errors that are being made.