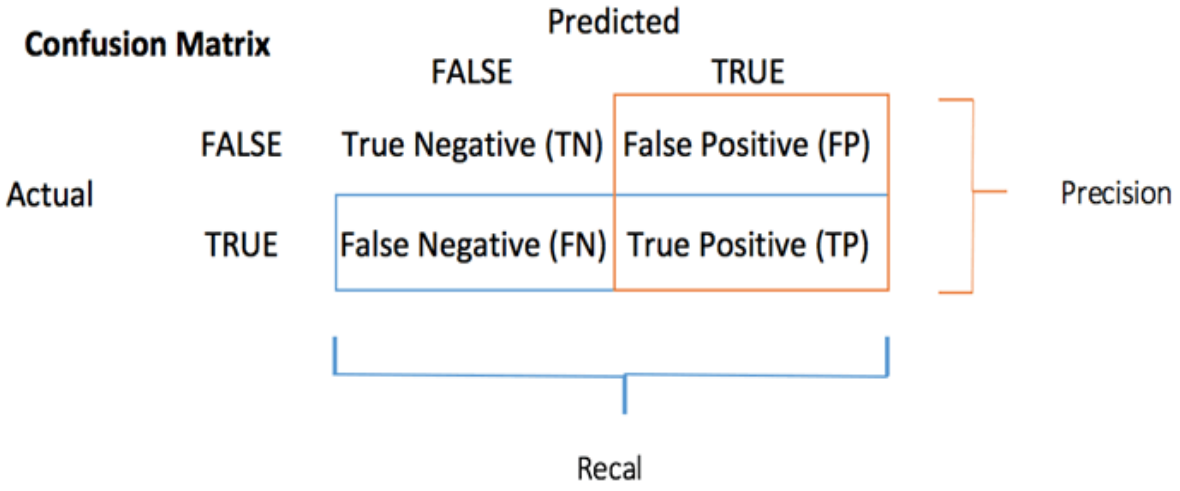
Lecture 13

Confusion matrix

In the field of machine learning and specifically the problem of statistical classification, a **confusion matrix**, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a **matching matrix**). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa)

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:



Confusion Table

TP: True Positive: Predicted values correctly predicted as actual 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.

FP: False Positive: Predicted values incorrectly predicted an actual positive. i.e., Negative values predicted as positive. Your prediction is positive, and it is false.

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

FN: False Negative: Positive values predicted as negative. Your prediction is negative, and result it is positive.

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

TN: True Negative: Predicted values correctly predicted as an actual negative. When you predicted negative, and it's true.

You had predicted that England would not win and it lost.

Example

let's focus in **binary classifiers** as with the spam filtering example, in which each email can be either spam or not spam. The confusion matrix will be of the following form:

The predicted classes are represented in the columns of the matrix, whereas the actual classes are in the rows of the matrix. We then have four cases:

- **True positives (TP)**: the cases for which the classifier predicted 'spam' and the emails were actually spam.
- **True negatives (TN)**: the cases for which the classifier predicted 'not spam' and the emails were actually real.
- **False positives (FP)**: the cases for which the classifier predicted 'spam' but the emails were actually real.
- **False negatives (FN)**: the cases for which the classifier predicted 'not spam' but the emails were actually spam.

In order to avoid confusion, note the following. 'True' or 'false' indicate if the classifier predicted the class correctly, whereas 'positive' or 'negative' indicate if the classifier predicted the desired class (in this case, 'positive' correspond to 'spam', as this is the type of email we want to predict).

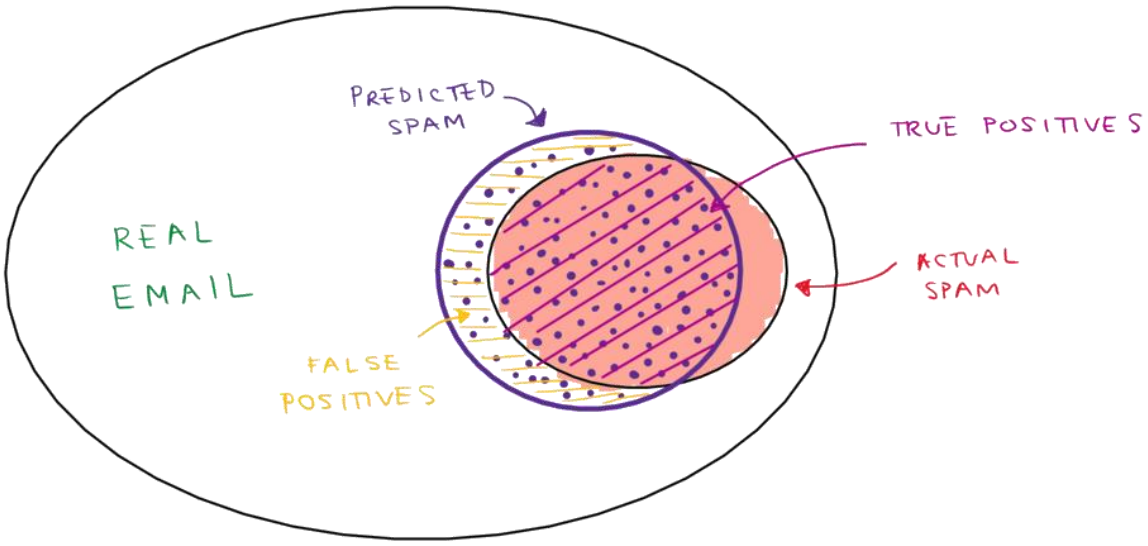


Diagram illustrating TP and FPs. The TP rate is the proportion of emails predicted as spam which are actually spam. The FP rate is the proportion of actual real emails which are predicted as spam.

Accuracy

The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

We obtain again that 99% of the predicted outputs were correctly classified. However, the confusion matrix allows us to have a better picture of the performance of the algorithm.

Precision, recall and f1-score

Besides the accuracy, there are several other performance measures which can be computed from the confusion matrix.

Let's go through the list:

Precision/ Positive Predictive value (PPV)

The precision metric shows the accuracy of the positive class. It measures how likely the prediction of the positive class is correct.

It answers the question:

"When it predicts the positive result, how often is it correct?"

This is obtained by using the following formulae:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Precision is usually used when the goal is to *limit the number of false positives* (FP). For example, this would be the metric to focus on if our goal with the spam filtering algorithm is to minimize the number of reals emails that are classified as spam.

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.

Recall/ Sensitivity:

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

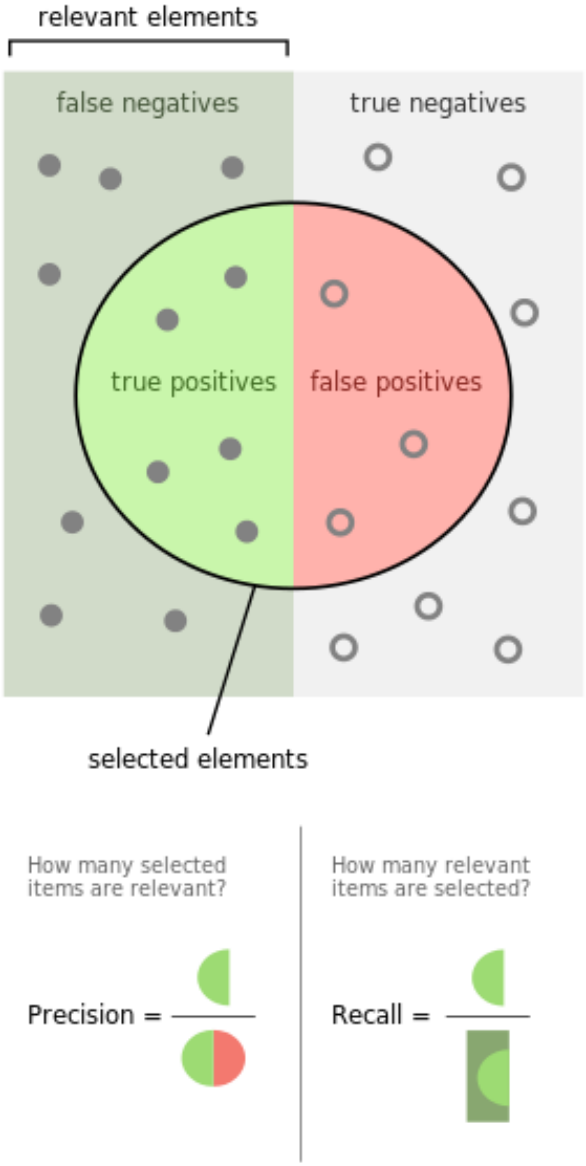
it answers the question:

"When it is actually the positive result, how often does it predict correctly?"

This is obtained by using the following formulae:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Recall is usually used when the goal is to *limit the number of false negatives* (FN). In our example, that would correspond to minimizing the number of spam emails that are classified as real emails. Recall is also known as "sensitivity" and "true positive rate" (TPR).



F1-score

F-score or **F-measure** is a measure of a test's accuracy. It is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive.

This is just the harmonic mean of precision and recall:

$$F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{FN + FP}{2}}$$

It is useful when you need to take both precision and recall into account. If you try to only optimize recall, your algorithm will predict most examples to belong to the positive class, but that will result in many false positives and, hence, low precision. On the other hand, if you try to optimize precision, your model will predict very few examples as positive results (the ones which highest probability), but recall will be very low.
Higher the F1-score, the better will be the predictive power of the classification procedure. The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero.

ROC curve

A more visual way to measure the performance of a binary classifier is the **receiver operating characteristic (ROC) curve**. It is created by plotting the true positive rate (TPR) (or recall) against the false positive rate (FPR), which we haven't defined explicitly yet:

So, false positive rate (FPR) is,

$$\text{FP rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}.$$

It is equal to 1 – the true negative rate (TNR), which is the ratio of negative instances that are correctly classified as negative.
The TNR is also called specificity. Hence, the ROC curve plots sensitivity (recall) versus 1 – specificity.

It also demonstrates a trade-off between sensitivity (recall) and specificity (the true negative rate).

$$\text{TN rate} = TN/(TN + FP)$$

The question it answers is the following:

"When it is actually the negative result, how often does it predict incorrectly?"
The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.

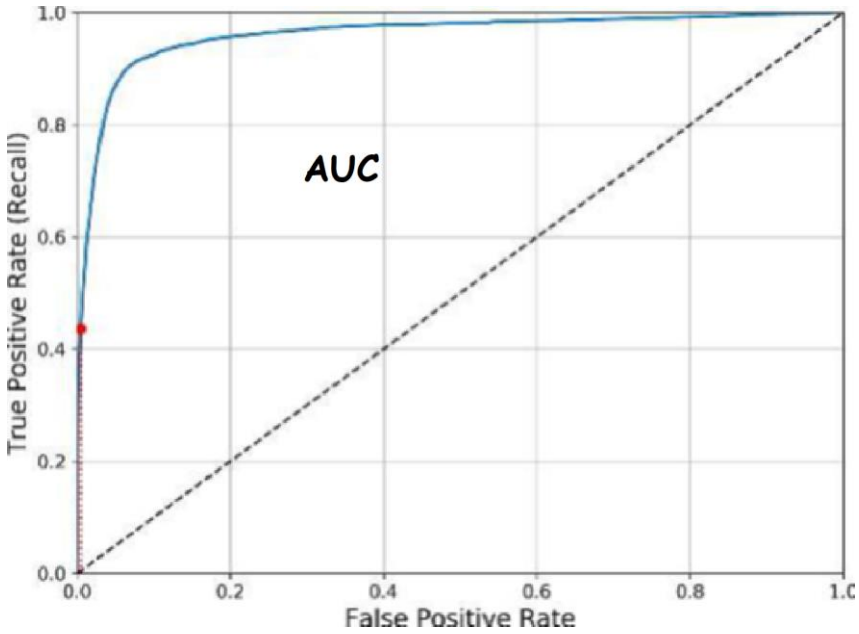


Figure. This ROC curve plots the false positive rate against the true positive rate for all possible thresholds

Once again there is a trade-off: the higher the recall (TPR), the more false positives (FPR) the classifier produces. The dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).

Area under the curve

When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative').