data school

Data science for beginners!

About

Most Popular Posts

Data Science Resources

Join my 9,000+ YouTube subscribers

Subscribe to Newsletter

Name:

Email:

Subscribe

March 26, 2014 · MACHINE LEARNING POPULAR

Simple guide to confusion matrix terminology

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

I wanted to create a **"quick reference guide" for confusion matrix terminology** because I couldn't find an existing resource that suited my requirements: compact in presentation, using numbers instead of arbitrary variables, and explained both in terms of formulas and sentences.

Let's start with an **example confusion matrix for a binary classifier** (though it can easily be extended to the case of more than two classes):

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

What can we learn from this matrix?

- There are two possible predicted classes: "yes" and "no". If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.
- The classifier made a total of 165 predictions (e.g., 165 patients were being tested for the presence of that disease).
- Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
- In reality, 105 patients in the sample have the disease, and 60 patients do not.

Let's now define the most basic terms, which are whole numbers (not rates):

- **true positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives (TN):** We predicted no, and they don't have the disease.
- **false positives (FP):** We predicted yes, but they don't actually have the

disease. (Also known as a "Type I error.")

- **false negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

I've added these terms to the confusion matrix, and also added the row and column totals:

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

This is a list of rates that are often computed from a confusion matrix:

- **Accuracy:** Overall, how often is the classifier correct?
  - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/105 = 0.95
  - also known as "Sensitivity" or "Recall"

- **False Positive Rate:** When it's actually no, how often does it predict yes?
    - FP/actual no = 10/60 = 0.17
- **Specificity:** When it's actually no, how often does it predict no?
    - TN/actual no = 50/60 = 0.83
    - equivalent to 1 minus False Positive Rate
- **Precision:** When it predicts yes, how often is it correct?
    - TP/predicted yes = 100/110 = 0.91
- **Prevalence:** How often does the yes condition actually occur in our sample?
    - actual yes/total = 105/165 = 0.64

A couple other terms are also worth mentioning:

- **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. (**More details about PPV.**)
- **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, as demonstrated by the **Accuracy Paradox**.

- **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. (**More details about Cohen's Kappa.**)
- **F Score:** This is a weighted average of the true positive rate (recall) and precision. (**More details about the F Score.**)
- **ROC Curve:** This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class. (**More details about ROC Curves.**)

And finally, for those of you from the world of Bayesian statistics, here's a quick summary of these terms from **Applied Predictive Modeling**:

> In relation to Bayesian statistics, the sensitivity and specificity are

> the conditional probabilities, the prevalence is the prior, and the positive/negative predicted values are the posterior probabilities.

What did I miss? Are there any terms that need a better explanation? Your feedback is welcome!

P.S. Want to Tweet about this post? **Here's a Tweet you can RT**.

Comments Community 1 Login

Recommend 4 Sort by Best

Join the discussion…

Eshwar Bandaru • 22 days ago

Thanks a lot for the simple yet wonderful article.

I'm confused with True positive rate and Precision. Which is a better metric for evaluating the correctness of the model when the actual value is "YES"? I believe it's the True positive rate, as it evaluates the accuracy with the actual value. When exactly to use each of these and If possible, Could you share the use of these metrics with an example?

• Reply • Share ›

Kevin Markham Mod

Eshwar Bandaru • 16 days ago

Great question, Eshwar. Neither

Great question, Eshwar. Neither metric is inherently "better", rather they serve different goals. 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).

Hope that helps!

Reply • Share ›

Venkateshwaralu Srikarunyan
• a month ago

a fantastic effort, thank you so much for this wonderful article :)

Reply • Share ›

Kevin Markham Mod
Venkateshwaralu Srikarunyan
• a month ago

You're welcome!

Reply • Share ›

rick davies • 3 months ago

It would be great if you could expand a bit more on how the contents of a Confusion Matrix can be interpreted in terms of Baysenian stats, which I just trying to get my head around. E.g. by an example calculation that links parts of the question to the cells of the Confusion Matrix

Reply • Share ›

apna • 4 months ago

how can i use it in data mining????

how can I use it in data mining????

Reply • Share ›

Kevin Markham Mod ➦ apna
• 3 months ago

Any of the metrics mentioned above may be useful for measuring the performance of a classification model. It all depends on which metric best represents your business objectives.

Reply • Share ›

Dheeb bashish • 4 months ago

Thank you for this excellent explanation, my question is if I have multi class and I want to computer AUC, area under ROC. Suppose I have 3 class this mean I will have also 3 AUC. So, which one I will choose for my system

Reply • Share ›

Kevin Markham Mod ➦
Dheeb bashish • 4 months ago

That's correct: If you have a classification problem with three classes, you would have 3 ROC curves (and thus 3 AUC scores). More information is here: http://www.dataschool.io/roc-c...

You may want to consider using a different evaluation metric instead... ROC/AUC is most suitable for two-class problems.

Reply • Share ›

Dheeb bashish ➦
Kevin Markham
• 4 months ago

Dear Markham,,,

Thank you,,,

However, I still think the AUC is better than

accuracy and Error rate.

in AUC multiclass, I found this paper is good "An introduction to ROC analysis"

http://www.sciencedirect.cc

they take the summation of all AUC * P(ci);
Thank you

▲ | ▼ • Reply • Share ›

Kevin Markham Mod ➦ Dheeb bashish • 3 months ago

Thanks for sharing, I look forward to reading that paper!

▲ | ▼ • Reply • Share ›

ajay kumar • 4 months ago

Respected Sir,
I have two confusion matrices and I want to perform McNemar Test. It is hereby to requesting you please tell me how to generate the values of 2 by 2 matrix I means how to find the values of f11, f12, f21 and f22 from the confusion matrices.

Thank You,

▲ | ▼ • Reply • Share ›

Kevin Markham Mod ➦ ajay kumar • 4 months ago

I'm not familiar with McNemar's

© 2015 Data School. All rights reserved. Powered by Ghost. Crisp theme by Kathy Qian.