CrossValidated

Why is accuracy not the best measure for assessing classification models?

Asked 7 years, 9 months ago Modified 5 months ago Viewed 143k times

255

This is a general question that was asked indirectly multiple times in here, but it lacks a single authoritative answer. It would be great to have a detailed answer to this for the reference.

Accuracy, the proportion of correct classifications among all classifications, is very simple and very "intuitive" measure, yet it may be a poor measure for imbalanced data. Why does our intuition misguide us here and are there any other problems with this measure?

machine-learning model-evaluation accuracy scoring-rules faq

Share Cite

Improve this question Follow

edited Apr 17, 2021 at 21:01

Sycorax ♦

95.4k 23 243 400

asked Nov 9, 2017 at 7:32

Tim

144k 26 275 527

2 I have posted a follow-up question asking for *peer-reviewed* references on this because, let's face it, a *peer-reviewed* publication gets more respect than even the most highly-voted CV thread: Academic reference on the drawbacks of accuracy, F1 score, sensitivity and/or specificity. – Stephan Kolassa Jan 30, 2023 at 16:26

12 Answers

Sorted by:

249

+50

Most of the other answers focus on the example of unbalanced classes. Yes, this is important. However, I argue that accuracy is problematic even with balanced classes.

Frank Harrell has written about this on his blog: Classification vs. Prediction and Damage Caused by Classification Accuracy and Other Discontinuous Improper Accuracy Scoring Rules.

Essentially, his argument is that the statistical component of your exercise ends when you output a probability for each class of your new sample. Mapping these predicted probabilities $(\hat{p}, 1-\hat{p})$ to a 0-1 classification, by choosing a threshold beyond which you classify a new observation as 1 vs. 0 is not part of the *statistics* any more. It is part of the *decision* component. And here, you need the probabilistic output of your model - but also considerations like:

- What are the consequences of deciding to treat a new observation as class 1 vs. 0? Do I then send out a cheap marketing mail to all 1s? Or do I apply an invasive cancer treatment with big side effects?
- What are the consequences of treating a "true" 0 as 1, and vice versa? Will I tick off a customer? Subject someone to unnecessary medical treatment?
- Are my "classes" truly discrete? Or is there actually a continuum (e.g., blood pressure), where clinical thresholds are in reality just cognitive shortcuts? If so, how *far* beyond a threshold is the case I'm "classifying" right now?
- Or does a low-but-positive probability to be class 1 actually mean "get more data", "run another test"?

Depending on the *consequences* of your decision, you will use a different threshold to make the decision. If the action is invasive surgery, you will require a much higher probability for your classification of the patient as suffering from something than if the action is to recommend two aspirin. Or you might even have *three* different decisions although there are only *two* classes (sick vs. healthy): "go home and don't worry" vs. "run another test because the one we have is inconclusive" vs. "operate immediately".

The correct way of assessing predicted probabilities $(\hat{p}, 1-\hat{p})$ is *not* to compare them to a threshold, map them to $(0, 1)$ based on the threshold and then assess the transformed $(0, 1)$ classification. Instead, one should use proper **scoring-rules**. These are loss functions that map predicted probabilities and corresponding observed outcomes to loss values, which are minimized in expectation by the true probabilities $(p, 1-p)$. The idea is that we take the average over the scoring rule evaluated on multiple (best: many) observed outcomes and the corresponding predicted class membership probabilities, as an estimate of the expectation of the scoring rule.

Note that "proper" here has a precisely defined meaning - there are *improper scoring rules* as well as *proper scoring rules* and finally *strictly proper scoring rules*. *Scoring rules* as such are loss functions of predictive densities and outcomes. *Proper scoring rules* are

scoring rules that are minimized in expectation if the predictive density is the true density. *Strictly proper scoring rules* are scoring rules that are *only* minimized in expectation if the predictive density is the true density.

As Frank Harrell notes, accuracy is an improper scoring rule. (More precisely, *accuracy is not even a scoring rule at all*: see my answer to Is accuracy an improper scoring rule in a binary classification setting?) This can be seen, e.g., if we have no predictors at all and just a flip of an unfair coin with probabilities $(0.6, 0.4)$. Accuracy is maximized if we classify everything as the first class and completely ignore the 40% probability that any outcome might be in the second class. *(Here we see that accuracy is problematic even for balanced classes.)* Proper **scoring-rules** will prefer a $(0.6, 0.4)$ prediction to the $(1, 0)$ one in expectation. In particular, accuracy is discontinuous in the threshold: moving the threshold a tiny little bit may make one (or multiple) predictions change classes and change the entire accuracy by a discrete amount. This makes little sense.

More information can be found at Frank's two blog posts linked to above, as well as in Chapter 10 of Frank Harrell's *Regression Modeling Strategies*.

(This is shamelessly cribbed from an earlier answer of mine.)

EDIT. My answer to Example when using accuracy as an outcome measure will lead to a wrong conclusion gives a hopefully illustrative example where maximizing accuracy can lead to wrong decisions *even for balanced classes*.

Share Cite Improve this answer Follow

edited Oct 13, 2022 at 2:28 — R.M. — 1,099 — 7 — 19

answered Nov 9, 2017 at 8:28 — Stephan Kolassa — 137k — 22 — 273 — 532

12 @Tim Frank's point (that he discussed in numerous answers on our site and elsewhere), as I understand it, is that if a classification algorithm does not return probabilities then it's garbage and should not be used. To be honest, most of the commonly used algorithms do return probabilities. – amoeba Nov 9, 2017 at 9:17

15 I'd say that an algorithm that takes past observations and outputs only classifications without taking the points above into account (e.g., costs of mis-decisions) conflates the statistical and the decision aspect. It's like someone recommending a particular type of car to you without first asking you whether you want to transport a little league baseball team, a bunch of building materials, or only yourself. So I'd also say such an algorithm would be garbage. – Stephan Kolassa Nov 9, 2017 at 9:23

15 I was going to write an answer, but then didn't need to. Bravo. I discuss this with my students as a "separation of concerns" between statistical modeling and decision making. This type of concept is very deeply rooted in engineering culture. – Matthew Drury Nov 9, 2017 at 14:34

10 @chainD: if your classifier (remember, it's the one with the *highest accuracy*) says that "everyone in this sample is healthy", then what doctor or analyst would believe that there is more to the story? I agree that in the end, it's a call for the analyst to make, but "everyone is healthy" is far less helpful to the analyst than something that draws attention to residual uncertainty like the 95%/5% prediction. – Stephan Kolassa Nov 10, 2017 at 20:43

19 @StephanKolassa 's answer and comments are superb. Someone else comment implied that there is a difference in how this is viewed depending on which culture you are part of. This is not

112

When we use accuracy, we assign equal cost to false positives and false negatives. When that data set is imbalanced - say it has 99% of instances in one class and only 1 % in the other - there is a great way to lower the cost. Predict that every instance belongs to the majority class, get accuracy of 99% and go home early.

The problem starts when the actual costs that we assign to every error are not equal. If we deal with a rare but fatal disease, the cost of failing to diagnose the disease of a sick person is much higher than the cost of sending a healthy person to more tests.

In general, there is no general best measure. The best measure is derived from your needs. In a sense, it is not a machine learning question, but a business question. It is common that two people will use the same data set but will choose different metrics due to different goals.

Accuracy is a great metric. Actually, most metrics are great and I like to evaluate many metrics. However, at some point you will need to decide between using model A or B. There you should use a single metric that best fits your need.

For extra credit, choose this metric before the analysis, so you won't be distracted when making the decision.

Share Cite Improve this answer Follow

edited Nov 13, 2017 at 6:52

answered Nov 9, 2017 at 7:45

DaL's DaL user **4,742** 3 21 28

4 Great answer - I've proposed a couple of edits just to try and make the point clearer to beginners in machine learning (at whom this question is aimed). – nekomatic Nov 9, 2017 at 8:47

1 I'd disagree that it's not a machine learning problem. But addressing it would involve doing machine learning on the meta problem and necessitate the machine having access to some kind of data beyond just the basic classification information. – Shufflepants Nov 9, 2017 at 20:41

3 I don't see it as a function of only the data since different goals can lad to different cost/model/performance/metrics. I do agree that in general, the question of cost can be handled mathematically. However questions like the cost of treating patients rely on totally different information. This information needed for the meta data is usually not suitable for machine learning methodology so most of the time it is handled with different methods. – DaL Nov 12, 2017 at 12:01

2 By "misdiagnosing a person with the disease", you mean "misdiagnosing a person *who has* the disease (as not having the disease)", right? Because that phrase could be interpreted either way. – Sophie Swett Nov 13, 2017 at 1:39

1 Accuracy could be adjusted for prevalence (*a-priori* class membership probability) - regardless of whether the data at hand is balanced or imbalanced, and whether the imbalanced data reflects prevalence for a given use-case/scenario or not. Just as predictive values should not be calculated on "raw" numbers of test cases *unless* the frequencies of the different classes properly reflect class prevalences for the application at hand. – cbeleites Nov 19, 2017 at 15:43

35

The problem with accuracy

Standard accuracy is defined as the ratio of correct classifications to the number of classifications done.

$$accuracy := \frac{\text{correct classifications}}{\text{number of classifications}}$$

It is thus an overall measure over all classes and as we'll shortly see it's not a good measure to tell an oracle apart from an actual useful test. An oracle is a classification function that returns a random guess for each sample. Likewise, we want to be able to rate the classification performance of our classification function. Accuracy *can* be a useful measure if we have the same amount of samples per class but if we have an imbalanced set of samples accuracy isn't useful at all. Even more so, a test can have a high accuracy but actually perform worse than a test with a lower accuracy.

If we have a distribution of samples such that 90% of samples belong to class $\mathcal{A}$, 5% belonging to $\mathcal{B}$ and another 5% belonging to $\mathcal{C}$ then the following classification function will have an accuracy of 0.9:

$$classify(sample) := \{\, \mathcal{A} \quad \text{if } \top$$

Yet, it is obvious given that we know how $classify$ works that this it can not tell the classes apart at all. Likewise, we can construct a classification function

$$classify(sample) := \text{guess} \begin{cases} \mathcal{A} & \text{with p } = 0.96 \\ \mathcal{B} & \text{with p } = 0.02 \\ \mathcal{C} & \text{with p } = 0.02 \end{cases}$$

which has an accuracy of $0.96 \cdot 0.9 + 0.02 \cdot 0.05 \cdot 2 = 0.866$ and will not always predict $\mathcal{A}$ but still given that we know how $classify$ works it is obvious that it can not tell classes apart. Accuracy in this case only tells us how good our classification function is at guessing. This means that accuracy is not a good measure to tell an oracle apart from a useful test.

Accuracy per Class

We can compute the accuracy individually per class by giving our classification function only samples from the same class and remember and count the number of correct classifications and incorrect classifications then compute $accuracy := \text{correct}/(\text{correct} + \text{incorrect})$. We repeat this for every class. If we have a classification function that can accurately recognize class $\mathcal{A}$ but will output a random guess for the other classes then this results in an accuracy of 1.00 for $\mathcal{A}$ and an accuracy of 0.33 for the other classes. This already provides us a much better way to judge the performance of our classification function. An oracle always guessing the same class will produce a per class accuracy of 1.00 for that class, but 0.00 for the other class. If our test is useful all the accuracies per class should be > 0.5. Otherwise, our test isn't

If our test is useful all the accuracies per class should be > 0.5. Otherwise, our test isn't better than chance. However, accuracy per class does not take into account false positives. Even though our classification function has a 100% accuracy for class $\mathcal{A}$ there will also be false positives for $\mathcal{A}$ (such as a $\mathcal{B}$ wrongly classified as a $\mathcal{A}$).

Sensitivity and Specificity

In medical tests sensitivity is defined as the ratio between people correctly identified as having the disease and the amount of people actually having the disease. Specificity is defined as the ratio between people correctly identified as healthy and the amount of people that are actually healthy. The amount of people actually having the disease is the amount of true positive test results plus the amount of false negative test results. The amount of actually healthy people is the amount of true negative test results plus the amount of false positive test results.

Binary Classification

In binary classification problems there are two classes $\mathcal{P}$ and $\mathcal{N}$. T_n refers to the number of samples that were correctly identified as belonging to class n and F_n refers to the number of samples that werey falsely identified as belonging to class n. In this case sensitivity and specificity are defined as following:

$$sensitivity := \frac{T_{\mathcal{P}}}{T_{\mathcal{P}} + F_{\mathcal{N}}}$$
$$specificity := \frac{T_{\mathcal{N}}}{T_{\mathcal{N}} + F_{\mathcal{P}}}$$

$T_{\mathcal{P}}$ being the true positives $F_{\mathcal{N}}$ being the false negatives, $T_{\mathcal{N}}$ being the true negatives and $F_{\mathcal{P}}$ being the false positives. However, thinking in terms of negatives and positives is fine for medical tests but in order to get a better intuition we should not think in terms of negatives and positives but in generic classes α and β. Then, we can say that the amount of samples correctly identified as belonging to α is T_α and the amount of samples that actually belong to α is $T_\alpha + F_\beta$. The amount of samples correctly identified as not belonging to α is T_β and the amount of samples actually not belonging to α is $T_\beta + F_\alpha$. This gives us the sensitivity and specificity for α but we can also apply the same thing to the class β. The amount of samples correctly identified as belonging to β is T_β and the amount of samples actually belonging to β is $T_\beta + F_\alpha$. The amount of samples correctly identified as not belonging to β is T_α and the amount of samples actually not belonging to β is $T_\alpha + F_\beta$. We thus get a sensitivity and specificity per class:

$$sensitivity_\alpha := \frac{T_\alpha}{T_\alpha + F_\beta}$$
$$specificity_\alpha := \frac{T_\beta}{T_\beta + F_\alpha}$$
$$\dots \quad \dots \qquad \frac{T_\beta}{}$$

$$sensitivity_\beta := \frac{T_\beta}{T_\beta + F_\alpha}$$
$$specificity_\beta := \frac{T_\alpha}{T_\alpha + F_\beta}$$

We however observe that $sensitivity_\alpha = specificity_\beta$ and $specificity_\alpha = sensitivity_\beta$. This means that if we only have two classes we don't need sensitivity and specificity per class.

N-Ary Classification

Sensitivity and specificity per class isn't useful if we only have two classes, but we can extend it to multiple classes. Sensitivity and specificity is defined as:

$$\text{sensitivity} := \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$
$$\text{specificity} := \frac{\text{true negatives}}{\text{true negatives} + \text{false-positives}}$$

The true positives is simply T_n, the false negatives is simply $\sum_i(F_{n,i})$ and the false positives is simply $\sum_i(F_{i,n})$. Finding the true negatives is much harder but we can say that if we correctly classify something as belonging to a class different than n it counts as a true negative. This means we have at least $\sum_i(T_i) - T(n)$ true negatives. However, this aren't all true negatives. All the wrong classifications for a class different than n are also true negatives, because they correctly weren't identified as belonging to n. $\sum_i(\sum_k(F_{i,k}))$ represents all wrong classifications. From this we have to subtract the cases where the input class was n meaning we have to subtract the false negatives for n which is $\sum_i(F_{n,i})$ but we also have to subtract the false positives for n because they are false positives and not true negatives so we have to also subtract $\sum_i(F_{i,n})$ finally getting $\sum_i(T_i) - T(n) + \sum_i(\sum_k(F_{n,i})) - \sum_i(F_{n,i}) - \sum_i(F_{i,n})$. As a summary we have:

$$\text{true positives} := T_n$$
$$\text{true negatives} := \sum_i(T_i) - T(n) + \sum_i(\sum_k(F_{n,i})) - \sum_i(F_{n,i}) - \sum_i(F_{i,n})$$
$$\text{false positives} := \sum_i(F_{i,n})$$
$$\text{false negatives} := \sum_i(F_{n,i})$$

$$sensitivity(n) := \frac{T_n}{T_n + \sum_i(F_{n,i})}$$
$$specificity(n) := \frac{\sum_i(T_i) - T_n + \sum_i(\sum_k(F_{i,k})) - \sum_i(F_{n,i}) - \sum_i(F_{i,n})}{\sum_i(T_i) - T_n + \sum_i(\sum_k(F_{i,k})) - \sum_i(F_{n,i})}$$

Introducing Confidence

We define a $confidence^{\top}$ which is a measure of how confident we can be that the reply of our classification function is actually correct. $T_n + \sum_i(F_{i,n})$ are all cases where the classification function replied with n but only T_n of those are correct. We thus define

$$confidence^{\top}(n) := \frac{T_n}{T_n + \sum_i(F_{i,n})}$$

But can we also define a $confidence^{\perp}$ which is a measure of how confident we can be that if our classification function responds with a class different than n that it actually wasn't an n?

Well, we get $\sum_i(\sum_k(F_{i,k})) - \sum_i(F_{i,n}) + \sum_i(T_i) - T_n$ all of which are correct except $\sum_i(F_{n,i})$.Thus, we define

$$confidence^{\perp}(n) = \frac{\sum_i(\sum_k(F_{i,k})) - \sum_i(F_{i,n}) + \sum_i(T_i) - T_n - \sum_i(F_{n,i})}{\sum_i(\sum_k(F_{i,k})) - \sum_i(F_{i,n}) + \sum_i(T_i) - T_n}$$

Share Cite Improve this answer Follow

edited Nov 13, 2019 at 14:10 user2740 1,396 2 13 22

answered Nov 9, 2017 at 12:55 mroman 501 4 11

Can you please provide any example of calculating Mean Accuracy using confusion matrix. – Aadnan Farooq A Sep 27, 2018 at 1:17

You can find a more detailed description with examples here: mroman.ch/guides/sensspec.html – mroman Sep 27, 2018 at 14:40

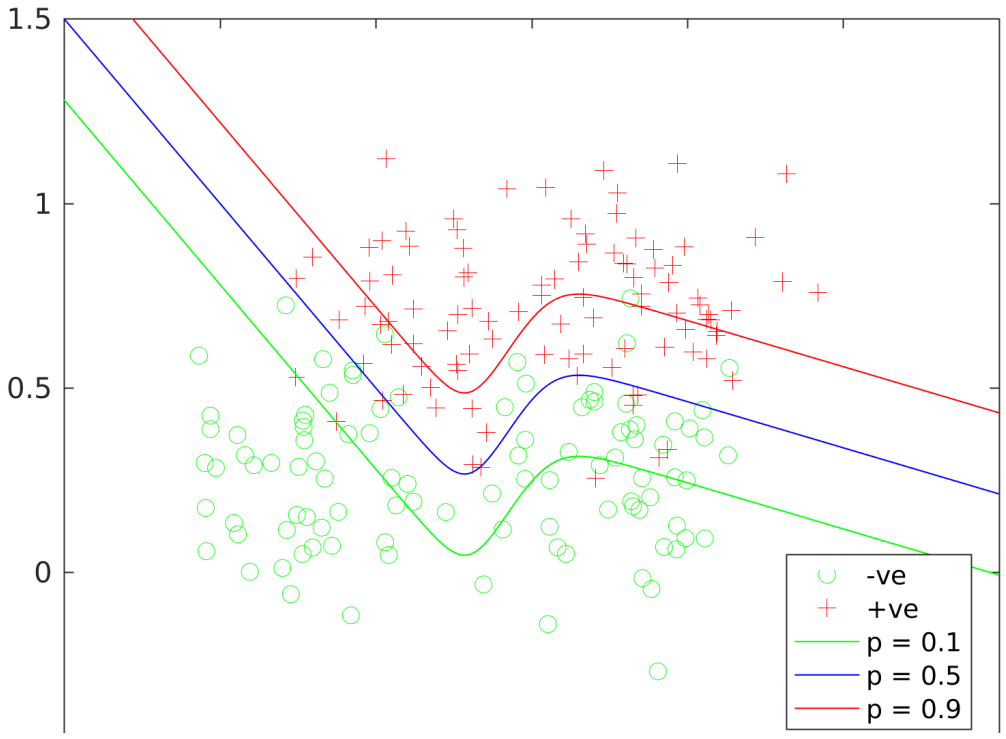
Reading through it again there's an error in the definition of confidence_false. I'm surprised nobody spotted that. I'll fix that in the next few days. – mroman Sep 27, 2018 at 16:14

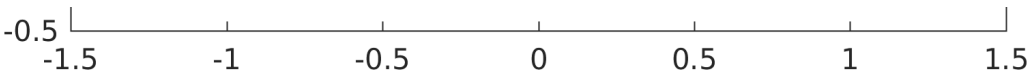
24

Here is a somewhat adversarial counter-example, where accuracy is better than a proper scoring rule, based on @Benoit_Sanchez's neat thought experiment (I have addressed the interpretability problem for imbalanced learning tasks in another answer),

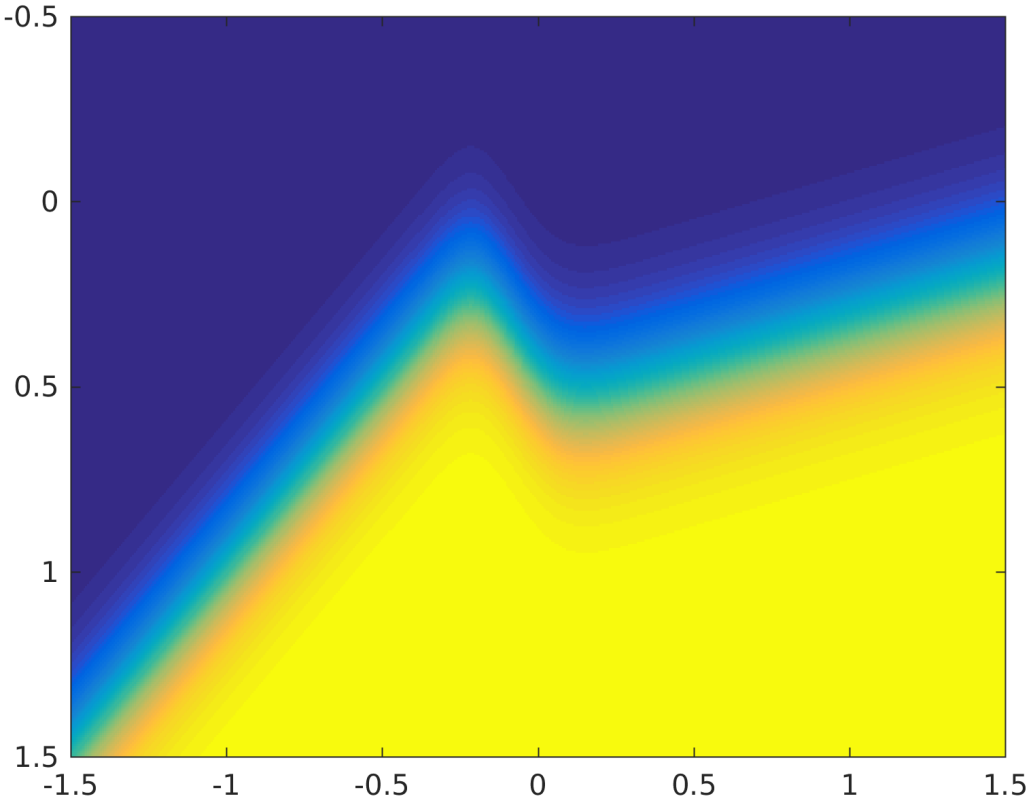
> You own an egg shop and each egg you sell generates a net revenue of 2 dollars. Each customer who enters the shop may either buy an egg or leave without buying any. For some customers you can decide to make a discount and you will only get 1 dollar revenue but then the customer will always buy.
>
> You plug a webcam that analyses the customer behaviour with features such as "sniffs the eggs", "holds a book with omelette recipes"... and classify them into "wants to buy at 2 dollars" (positive) and "wants to buy only at 1 dollar" (negative) before he leaves.
>
> If your classifier makes no mistake, then you get the maximum revenue you can expect. If it's not perfect, then:
>
> - for every false positive you lose 1 dollar because the customer leaves and you didn't try to make a successful discount
> - for every false negative you lose 1 dollar because you make a useless discount
>
> Then the accuracy of your classifier is exactly how close you are to the maximum revenue. It is the perfect measure.

So say we record the amount of time the customer spends "sniffing eggs" and "holding a book with omelette recipes" and make ourselves a classification task:

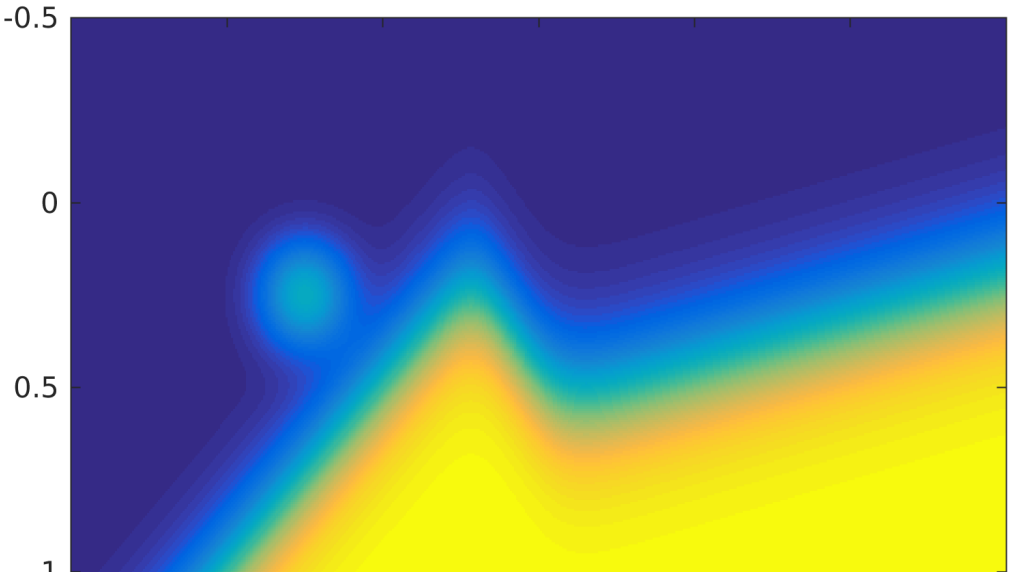


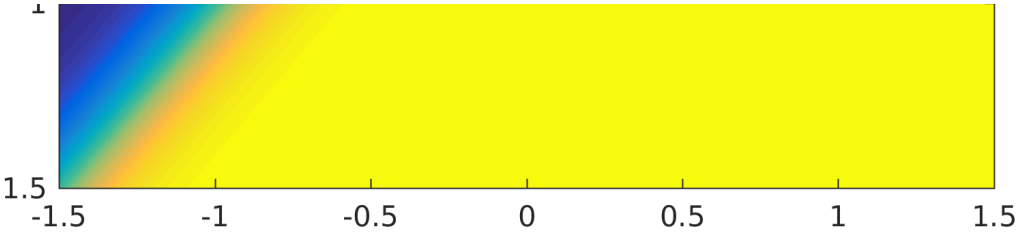


This is actually my version of Brian Ripley's synthetic benchmark dataset, but lets pretend it is the data for our task. As this is a synthetic task, I can work out the probabilities of class membership according to the true data generating process:

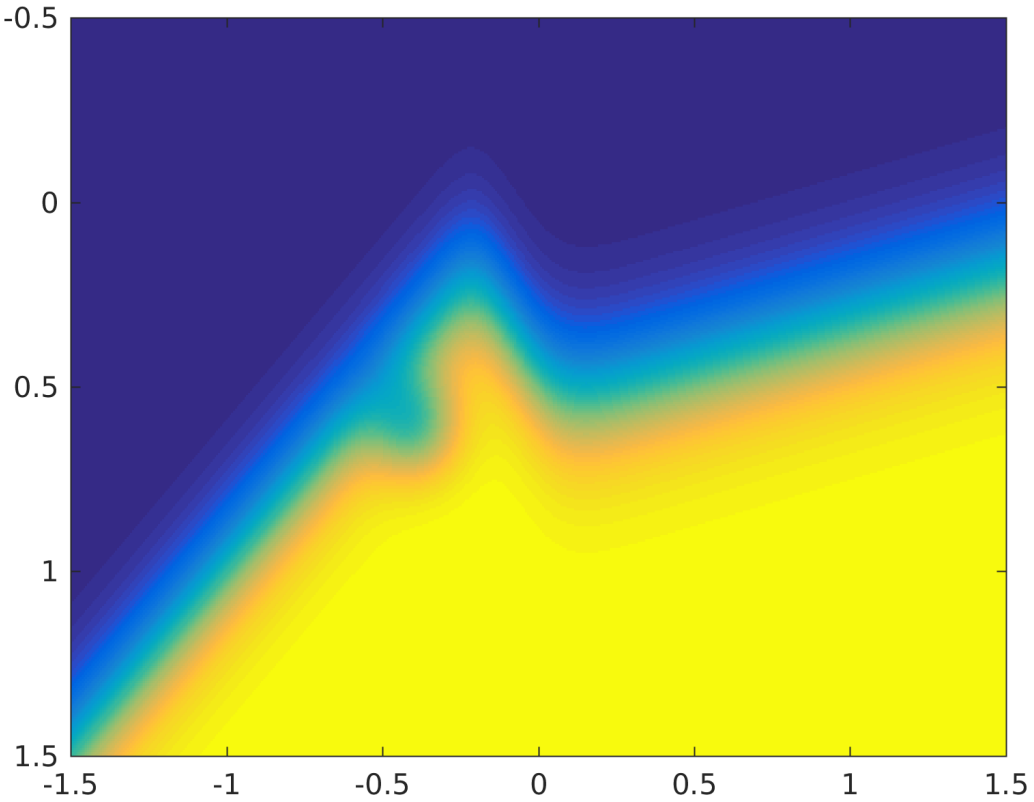


Unfortunately it is upside-down because I couldn't work out how to fix it in MATLAB, but please bear with me. Now in practice, we won't get a perfect model, so here is a model with an error (I have just perturbed the true posterior probabilities with a Gaussian bump).





And here is another one, with a bump in a different place.



Now the Brier score is a proper scoring rule, and it gives a slightly lower (better) score for the second model (because the perturbation is in a region of slightly lower density). However, the perturbation in the first model is well away from the decision boundary, and so that one has a higher accuracy.

Since in this particular application, the accuracy is equal to our financial gain in dollars, the Brier score is selecting the wrong model, and we will lose money.

Vapnik's advice that it is often better to form a purely discriminative classifier directly (rather than estimate a probability and threshold it) is based on this sort of situation. ***If*** all we are interested in is making a binary decision, then we don't really care what the classifier does away from the decision boundary, so we shouldn't waste resources modelling features of the data distribution that don't affect the decision.

This is a Laconic "if" though. If it is a classification task with fixed misclassification costs, no covariate shift and known and constant operational class priors, then this approach may indeed be better (and the success of the SVM in many practical applications is some

evidence of that). However, many applications are not like that, we may not know ahead of time what the misclassification costs are, or equivalently the operational class frequencies. In those applications we are much better off with a probabilistic classifier, and set the thresholds appropriately according to operational conditions.

Whether accuracy is a good performance metric depends on the needs of the application, there is no "one size fits all" policy. We need to understand the tools we use and be aware of their advantages and pitfalls, and consider the purpose of the exercise in choosing the right tool from the toolbox. In this example, the problem with the Brier score is that it ignores the true needs of the application, and no amount of adjusting the threshold will compensate for its selection of the wrong model.

It is also important to make a distinction between performance evaluation and model selection - they are not the same thing, and sometimes (often?) it is better to have a proper scoring rule for model selection in order to achieve maximum performance according to your metric of real interest (e.g. accuracy).

edited Mar 24 at 17:56

answered Jul 30, 2021 at 8:29

Dikran Marsupial
57.9k 9 154 236

2 "Why do you say this shows that accuracy picks the right model rather than accuracy sometimes picks the right model? " I wrote "Here is a somewhat adversarial counter-example, where accuracy is better than a proper scoring rule," which is clearly stating that this is one example where this is the case and is in no way stating a general rule. I even highlighted that it was an *adversarial* example. – Dikran Marsupial Aug 11, 2021 at 19:07

2 @Dave I have explained why this is important in the body of the answer. If you have an application where accuracy *IS* your primary concern, then deferring the classification can make performance worse by focusing modelling on aspects of the data that are unimportant. This is why SVMs have been so successful for that particular type of application. It is the insistence of a one-size-fits all approach that I think is wrong. It isn't accuracy **or** proper scoring rules - both have their uses. – Dikran Marsupial Aug 12, 2021 at 11:23

2 The important thing is understanding *WHY* in this case the Brier score picks the wrong model. Understand that, and it is easier to see why the SVM (and other purely discriminative classifiers) perform better in some applications (but worse in others) and shouldn't be discarded a-priori. – Dikran Marsupial Aug 12, 2021 at 11:25

2 A great answer! Related: "When is it appropriate to use an improper scoring rule?". @Dave, perhaps interesting for you, too. – Richard Hardy Jan 10, 2022 at 19:56

2 @RichardHardy the example in the accepted answer to that question is a very nice example of getting distracted by features of the data distribution that are irrelevant to the end goal of classification. Cheers! – Dikran Marsupial Jan 10, 2022 at 19:57

16

Imbalanced classes in your dataset

To be short: imagine, 99% of one class (say apples) and 1% of another class is in your data set (say bananas). My super duper algorithm gets an astonishing 99% accuracy for this data set, check it out:

```
return "it's an apple"
```

He will be right 99% of the time and therefore gets a 99% accuracy. Can I sell you my algorithm?

Solution: don't use an absolute measure (accuracy) but a relative-to-each-class measure (there are a lot out there, like ROC AUC)

Share Cite

Improve this answer Follow

edited Apr 4, 2019 at 9:47

NelsonGon

117 9

answered Nov 9, 2017 at 17:34

jonas-eschle

1,160 8 19

Nope, AUC is also not appropriate for imbalanced dataset. – SiXUlm Aug 16, 2019 at 21:39

@SiXUlm, can you elaborate on that? – jonas-eschle Aug 17, 2019 at 0:01

AUC is the area under ROC curve. The ROC curve is the plot of TPR vs FPR. Now, in the Bayesian setting, the imbalance is the odd of prior probability: $P(D)/P(D^C)$. The TPR can be seen as $P(T|D)$ and FPR can be seen as $P(F|D^C)$. The prior probability has nothing to do with the likelihood. – SiXUlm Aug 17, 2019 at 23:38

1 Hmm, sorry about that, I think you are correct. Actually I made a logical error and misinterpreted the effect of insensitivity to the performance. – SiXUlm Aug 19, 2019 at 20:10

7 AUC is actually appropriate for an imbalanced dataset, in fact it is one of the better metrics. In the extreme case; it is a relatively insensitive to false positives in the positive fraction compared with precision recall curves, but saying it is inappropiate for class imbalance is just incorrect. – Christopher John Oct 3, 2019 at 12:51

7

DaL answer is just exactly this. I'll illustrate it with a very simple example about... selling eggs.

You own an egg shop and each egg you sell generates a net revenue of 2 dollars. Each customer who enters the shop may either buy an egg or leave without buying any. For some customers you can decide to make a discount and you will only get 1 dollar revenue but then the customer will always buy.

You plug a webcam that analyses the customer behaviour with features such as "sniffs the eggs", "holds a book with omelette recipes"... and classify them into "wants to buy at 2 dollars" (positive) and "wants to buy only at 1 dollar" (negative) before he leaves.

If your classifier makes no mistake, then you get the maximum revenue you can expect. If it's not perfect, then:

- for every false positive you loose 1 dollar because the customer leaves and you didn't try to make a successful discount
- for every false negative you loose 1 dollar because you make a useless discount

Then the accuracy of your classifier is exactly how close you are to the maximum revenue. It is the perfect measure.

But now if the discount is a dollars. The costs are:

- false positive: a
- false negative: $2 - a$

Then you need an accuracy weighted with these numbers as a measure of efficiency of the classifier. If $a = 0.001$ for example, the measure is totally different. This situation is likely related to imbalanced data: few customers are ready to pay 2, while most would pay 0.001. You don't care getting many false positives to get a few more true positives. You can adjust the threshold of the classifier according to this.

If the classifier is about finding relevant documents in a database for example, then you can compare "how much" wasting time reading an irrelevant document is compared to finding a relevant document.

Share Cite Improve this answer Follow

answered Nov 9, 2017 at 17:40

Benoit Sanchez

8,947 29 52

3 This is an excellent answer. The performance metric depends on the requirements of the application, and sometimes the accuracy (or more generally the expected loss) is what we are interested in. Of course that doesn't mean it is necessarily the right metric for model selection (e.g. optimising hyper-parameters), but that doesn't mean it shouldn't be used for performance evaluation (or that the "class imbalance problem" is not a problem). – Dikran Marsupial Jul 27, 2021 at 8:32

This example seems to illustrate my point: it makes most sense to first make *probabilistic predictions* about the probability a given customer will purchase at a given discount, then make *decisions* on whether or not to offer a given discount to a given customer. The predictions are one input into the decisions, but there are also others (like the discount amount). Yes, in the end it may turn into a "weighted accuracy", but I'd say separating concerns is clearer. For one, a separation can easily be extended to multi-class problems. – Stephan Kolassa Jul 27, 2021 at 9:06

1 @StephanKolassa No, for the problem as stated, accuracy *is* the quantity of interest. Extending the scenario to include new features, such as variable discount amounts, is avoiding the point made by the analogy. I agree that it is a generally good idea to estimate the probability and then threshold at a suitable level, but the point remains that for this particular scenario, the accuracy *is* the quantity of interest. – Dikran Marsupial Jul 30, 2021 at 6:56

1 The work of Vapnik (especially the SVM) provides justification for solving the classification problem directly (avoids wasting resources, including data, on features that are irrelevant to the decision). For me, whether that is a good idea depends on the requirements of the application. It seems fine for e.g. handwritten digit recognition, where costs and priors are fixed, less so for things like medical diagnosis where operational conditions are more fluid. – Dikran Marsupial Jul 30, 2021 at 6:58

1 I disagree, it is the quantity of interest because it is the financial loss (up to a multiplicative constant). For the threshold argument, that only matters if it is too costly to retrain the classifier, and as Vapnik points out, a purely discriminative classifier may make better use of the data. I agree with both you and Frank Harrell *and* Vladimir Vapnik, but I don't think either is right all the time. It depends on the requirements of the application. – Dikran Marsupial Jul 30, 2021 at 7:03

5

After reading through all the answers above, here is an appeal to common sense. Optimality is a flexible term and always needs to be qualified; in other words, saying a model or algorithm is "optimal" is meaningless, especially in a scientific sense.

Whenever anyone says they are scientifically optimizing something, I recommend asking a question like: "In what sense do you define optimality?" This is because in science, unless you can measure something, you cannot optimize (maximize, minimize, etc.) it.

As an example, the OP asks the following:

"*Why is accuracy not the best measure for assessing classification models?*"

There is an embedded reference to optimization in the word "best" from the question above. "Best" is meaningless in science because "goodness" cannot be measured scientifically.

The scientifically correct response to this question is that the OP needed to define what "good" means. In the real world (outside of academic exercises and Kaggle competitions) there is always a cost/benefit structure to consider when using a machine to suggest or make decisions to or on behalf of/instead of people.

For classification tasks, that information can be embedded in a cost/benefit matrix with entries corresponding to those of the confusion matrix. Finally, since cost/benefit information is a function of the people who are considering using mechanistic help for their decision-making, it is subject to change with the circumstances, and therefore, there is never going to be one fixed measure of optimality which will work for all time in even one problem, let alone all problems (i.e., "models") involving classification.

Any measure of optimality for classification which ignores costs does so at its own risk. Even the ROC AUC fails to be cost-invariant, as shown in this figure.

Share Cite Improve this answer Follow

answered Mar 28, 2020 at 18:36

tdMJN6B2JtUe
396 3 13

3 One of my favorite quotes from a statistician (Youden): "It is, in fact, not a statistical matter to decide what weights should be attached to these two types of diagnostic error." First page of acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/... – tdMJN6B2JtUe Mar 28, 2020 at 19:00

Link to my full master's thesis (from which the above linked image was taken): researchgate.net/publication/... – tdMJN6B2JtUe Oct 14, 2020 at 21:18

After reading the answers above and discussions of "imbalanced classes," I am reminded of N. N. Taleb's writings (e.g., "The Black Swan"). Just sayin... – tdMJN6B2JtUe Nov 9, 2020 at 14:38

3

Classification accuracy is the number of correct predictions divided by the total number of predictions.

Accuracy can be misleading. For example, in a problem where there is a large class imbalance, a model can predict the value of the majority class for all predictions and achieve a high classification accuracy. So, further performance measures are needed such as F1 score and Brier score.

Share Cite Improve this answer Follow

answered Sep 27, 2018 at 14:27

jeza
2,129 4 28 45

3

I wrote a whole blog post on the matter: https://blog.ephorie.de/zeror-the-simplest-possible-classifier-or-why-high-accuracy-can-be-misleading

ZeroR, the simplest possible classifier, just takes the majority class as the prediction. With highly imbalanced data you will get a very high accuracy, yet if your minority class is the class of interest, this is completely useless. Please find the details and examples in the post.

Bottom line: when dealing with imbalanced data you can construct overly simple classifiers that give a high accuracy yet have no practical value whatsoever...

Share Cite Improve this answer Follow

edited Apr 28, 2020 at 12:08

answered Apr 28, 2020 at 10:06

vonjd
6,286 5 52 67

2 First, please expand this rather then giving link-only answer. Second, I don't agree with your conclusion, in many cases the data is imbalanced and it may have a lot of practical value, e.g. as discussed in this thread: stats.stackexchange.com/questions/283170/… – Tim Apr 28, 2020 at 12:01

@Tim: I don't see any practical value in ignoring the minority class completely (especially not when this is the class of interest!). – vonjd Apr 28, 2020 at 12:05

@Tim: Added more details and the gist of the post, hope this helps... Thank you! – vonjd Apr 28, 2020 at 12:09

@Tim:...oh, and thank you for the link to the other question, very interesting! – vonjd Apr 28, 2020 at 12:11

1 Thanks, it's clearer now. – Tim Apr 28, 2020 at 13:53

3

To address the original question more directly,

> 'Accuracy, the proportion of correct classifications among all classifications, is very simple and very "intuitive" measure'

This is indeed true, and sometimes it is better to have a metric that the "client" (if you have one) understands than a better metric that they don't. It also has the advantage that sometimes it is the metric of primary interest for the practical application (as mentioned in my previous answer).

> "yet it may be a poor measure for imbalanced data. Why does our intuition misguide us here and are there any other problems with this measure?"

This is also true, but there is an easy fix for our intuition failure. Rather than look at accuracy, consider the gain in accuracy that we can achieve by using the input data. For a binary classification problem, we could use something like:

$$score = \frac{Accuracy - \pi}{1 - \pi}$$

Where π is the prior probability (class frequency) of the majority class. Our intuition works much better with this score: A perfect classifier will give a score of 1; a classifier that assigns all patterns to the majority class ("just guessing") gets a score of 0. Any classifier that fails to be as good as guessing the majority class gets a negative score, which is nice and easy to understand.

This has been in use since at least the early 1990s (I first saw it on a training course I did as a student), and it is a special case of Cohen's kappa statistic.

The important thing here, is that this is just an affine transformation of accuracy, so it tells us *exactly* the same thing - it just does it on a scale that is less prone to misinterpretation by our intuitive biases. It is a bit like the difference between Centigrade, Farenheit and Kelvin - they are all temperature scales that tell us the same thing.

I think this *completely* solves the issue relating to imbalanced learning tasks.

Accuracy still has problems as a model selection criterion that are not solved by this rescaling (discontinuous, can be brittle), but we should make a distinction between model selection and model evaluation, and we don't need to use the same criterion for both (I often use Brier score for model selection even where accuracy is a key performance metric).

1 This and this also discuss the $score$ given above. // I think many issues related to performance metrics would be resolved by people using this instead of just accuracy. – Dave Mar 24 at 16:01

Share Cite Improve this answer Follow

edited Mar 24 at 17:46

answered Mar 24 at 15:51

Dikran Marsupial

57.9k 9 154 236

1

One of the problems of accuracy is that it ignores the intrinsic difficulty in the data-generating mechanism. Here, the difficulty refers to the uncertainty of the label, which can be measured by its variance. The point is that when assessing a classifier, a misclassification can be due to high uncertainty of the data-generating mechanism instead of the flaw of the model.

For instance, we predict $Y \in \{0, 1\}$ from $X \in \mathcal{X}$. Assume that there is an unknown data-generating probability function $P(Y = 1 \mid X = x)$, $x \in \mathcal{X}$. The variance

$$Var(Y = 1 \mid X = x) = P(Y = 1 \mid X = x)(1 - P(Y = 1 \mid X = x)),$$

which is maximized when $P(Y = 1 \mid X = x) = 0.5$ and minimized when $P(Y = 1 \mid X = x) = 0$ or 1. Suppose $\exists x_1, x_2 \in \mathcal{X}$ such that

$$P(Y = 1 \mid X = x) = \begin{cases} 0.5 & \text{when } x = x_1, \\ 0.99 & \text{when } x = x_2. \end{cases}$$

When evaluating the performance of a classifier, a misclassification at x_1 should be considered less severe than a misclassification at x_2. However, this problem of heteroscedasticity in variance is not reflected in accuracy. Also, the best possible accuracy depends on the $P(Y = 1 \mid X = x)$ and can be very different on different data sets.

To take the unknown $P(Y = 1 \mid X = x)$ into account and assess how well the classifier performance compared with the best possible performance, we may apply goodness-of-fit tests, e.g., Pearson's chi-squared test, Residual deviance test, and Hosmer–Lemeshow test. Although the majority of the goodness-of-fit tests only apply to parametric models, there is a recent work Is a Classification Procedure Good Enough?—A Goodness-of-Fit Assessment Tool for Classification Learning that addresses the evaluation problem of general classifiers.

1 What "intrinsic difficulty"? It's a big vague... – Tim Oct 20, 2023 at 7:07

Share Cite edited Oct 22, 2023 at 1:48 answered Oct 19, 2023 at 19:03

Improve this answer Follow

@Tim Sorry for the lack of clarification. The difficulty refers to the uncertainty of the label Y, which can be measured by its variance $P(Y=1 \mid X=x)(1-P(Y=1 \mid X=x))$. My point is that a misclassification can be due to the high uncertainty of Y, e.g., both $P(Y=1 \mid X=x)$ and $P(Y=0 \mid X=x)$ equal 0.5, instead of the problem of the classifier. Therefore, it would be ideal if we can also take this uncertainty into account when assessing a classifier. – augustin4 Oct 21, 2023 at 16:56

It's probably worth putting this in your answer. – Tim Oct 21, 2023 at 20:15

@Tim Thank you for your suggestion! I have made an update. – augustin4 Oct 22, 2023 at 1:50

-3

You may view accuracy as the R^2 of classification: an initially appealing metric with which to compare models, that falls short under detailed examination.

In both cases overfitting can be a major problem. Just as in the case of a high R^2 might mean that you are modelling the noise rather than the signal, a high accuracy may be a red-flag that your model applied too rigidly to your test dataset and does not have general applicability. This is especially problematic when you have highly imbalanced classification categories. The most accurate model might be a trivial one which classifies all data as one category (with the accuracy equal to proportion of the most frequent category), but this accuracy will fall spectacularly if you need to classify a dataset with a different true distribution of categories.

As others have noted, another problem with accuracy is an implicit indifference to the price of failure - i.e. an assumption that all mis-classifications are equal. In practice they are not, and the costs of getting the wrong classification is highly subject dependent and you may prefer to minimise a particular kind of wrongness than maximise accuracy.

Share Cite Improve this answer Follow

answered Nov 9, 2017 at 11:05

James
2,184 18 22

2 Hum. (1) I'd assume that evaluating accuracy or any other metric *out-of-sample* would be understood, so I don't really see how accuracy has more of a *specific overfitting problem*. (2) if you apply a model trained on population A to a *different* population B, then you are comparing apples to oranges, and I again don't really see how this is a *specific problem for accuracy*. – Stephan Kolassa Nov 9, 2017 at 11:28

1 (1) It is nevertheless a problem for accuracy, and the question is about using accuracy as a gold-standard. (2) The point of building a classifier is to use it on the oranges, not the just the apples. It should be general enough to capture the essential signals in the data (such that they exist), rather than being a catechism for your training data. – James Nov 9, 2017 at 11:45

Protected question. To answer this question, you need to have at least 10 reputation on this site (not counting the association bonus). The reputation requirement helps protect this question from spam and non-answer activity.

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

machine-learning model-evaluation accuracy scoring-rules faq

See similar questions with these tags.