




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

77  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.

58   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

classification

accuracy

model-evaluation

scoring-rules

share cite  
improve this question

edited Aug 3 '18 at 13:41



gung ♦

106k 34 258 520

asked Nov 9 '17 at 7:32



Tim ♦

55.6k 9 124 213


add a comment

7 Answers

active

oldest

votes

81  **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.**



+50

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  $(\hat{p}, 1 - \hat{p})$   $(p^{\wedge}, 1 - p^{\wedge})$  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  $(\hat{p}, 1 - \hat{p})$  ( $p^{\wedge}, 1 - p^{\wedge}$ ) is *not* to compare them to a threshold, map them to  $(0, 1)$  ( $0, 1$ ) based on the threshold and then assess the transformed  $(0, 1)$  ( $0, 1$ ) 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  $(p, 1 - p)$  ( $p, 1 - p$ ). 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  $(0.6, 0.4)$  ( $0.6, 0.4$ ). 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  $(0.6, 0.4)$  ( $0.6, 0.4$ ) prediction to the  $(1, 0)$  ( $1, 0$ ) 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 edited Oct 1 '18 at 6:22

answered Nov 9 '17 at 8:28



Stephan Kolassa

43.7k 6 90 160

- 6 @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 '17 at 9:17
- 5 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 '17 at 9:23
- 6 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 '17 at 14:34
- 6 @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 '17 at 20:43
- 11 @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 really the case; it's just that some fields bothered to understand the literature and others didn't. Weather forecasting, for example, has been at the forefront and has used proper scoring rules for assessing forecaster accuracy since at least 1951. – [Frank Harrell](#) Nov 11 '17 at 12:53

show 23 more comments



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](#) [edited Nov 13 '17 at 6:52](#)

answered Nov 9 '17 at 7:45



**DaL**

3,157 10 26

- 
- 3 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 '17 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 '17 at 20:41
- 
- 3 I don't see it as a function of only the data since different goals can lead 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 '17 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. – [Tanner Swett](#) Nov 13 '17 at 1:39
- 

You are right Tanner. I changed the test to make it clearer. – [DaL](#) Nov 13 '17 at 6:52

[show 4 more comments](#)



## The problem with accuracy

14

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

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

accuracy:=correct classificationsnumber 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 AA, 5% belonging to BB and another 5% belonging to CC then the following classification function will have an accuracy of 0.909:

$$\text{classify}(\text{sample}) := \begin{cases} A & \text{if } T \\ A & \text{if } T \end{cases}$$

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

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

classify(sample):=guess{Awith p =0.96Bwith p =0.02Cwith p =0.02

which has an accuracy of  $0.96 \cdot 0.9 + 0.02 \cdot 0.05 \cdot 2 = 0.866$   $0.96 \cdot 0.9 + 0.02 \cdot 0.05 \cdot 2 = 0.866$  and will not always predict AA 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

$$\text{accuracy} := \text{correct} / (\text{correct} + \text{incorrect}) \quad \text{accuracy} := \text{correct} / (\text{correct} + \text{incorrect})$$

. We repeat this for every class. If we have a classification function that can accurately recognize class AA but will output a random guess for the other classes then this results in an accuracy of 1.00 for AA 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 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 AA there will also be false positives for AA (such as a BB wrongly classified as a AA).

## 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 P and 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 were falsely identified as belonging to class n. In this case sensitivity and specificity are defined as following:

$$\text{sensitivity} := \frac{T_P}{T_P + F_N}$$

$$\text{specificity} := \frac{T_N}{T_N + F_P}$$

$$\text{sensitivity} = \frac{T_P}{T_P + F_N} \quad \text{specificity} = \frac{T_N}{T_N + F_P}$$

$T_P$  being the true positives  $F_N$  being the false negatives,  $T_N$  being the true negatives and  $F_P$  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  $\alpha$  and  $\beta$ . Then, we can say that the amount of samples correctly identified as belonging to  $\alpha$  is  $T_\alpha$  and the amount of samples that actually belong to  $\alpha$  is  $T_\alpha + F_\beta$ . The amount of samples correctly identified as not belonging to  $\alpha$  is  $T_\beta$  and the amount of samples actually not belonging to  $\alpha$  is  $T_\beta + F_\alpha$ . This gives us the sensitivity and specificity for  $\alpha$  but we can also apply the same thing to the class  $\beta$ . The amount of samples correctly identified as belonging to  $\beta$  is  $T_\beta$  and the amount of samples actually belonging to  $\beta$  is  $T_\beta + F_\alpha$ . The amount of samples correctly identified as not belonging to  $\beta$  is  $T_\alpha$  and the amount of samples actually not belonging to  $\beta$  is  $T_\alpha + F_\beta$ . We thus get a sensitivity and specificity per class:

$$\text{sensitivity}_\alpha := \frac{T_\alpha}{T_\alpha + F_\beta}$$

$$\text{specificity}_\alpha := \frac{T_\beta}{T_\beta + F_\alpha}$$

$$\text{sensitivity}_\beta := \frac{T_\beta}{T_\beta + F_\alpha}$$

$$\text{specificity}_\beta := \frac{T_\alpha}{T_\alpha + F_\beta}$$

$$\text{sensitivity}_\alpha := \frac{T_\alpha}{T_\alpha + F_\beta} \quad \text{specificity}_\alpha := \frac{T_\beta}{T_\beta + F_\alpha} \quad \text{sensitivity}_\beta := \frac{T_\beta}{T_\beta + F_\alpha} \quad \text{specificity}_\beta := \frac{T_\alpha}{T_\alpha + F_\beta}$$

We however observe that

$$\text{sensitivity}_\alpha = \text{specificity}_\beta \quad \text{sensitivity}_\alpha = \text{specificity}_\beta \text{ and}$$

$$\text{specificity}_\alpha = \text{sensitivity}_\beta \quad \text{specificity}_\alpha = \text{sensitivity}_\beta.$$

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:

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

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

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

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  $n$ .

$\sum_i (\sum_k (F_{i,k})) - \sum_i (F_{n,i})$  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

$$\sum_i (F_{i,n}) \text{ finally getting } \sum_i (T_i) - T(n) + \sum_i (\sum_k (F_{n,i})) - \sum_i (F_{n,i}) - \sum_i (F_{i,n}) = \sum_i (T_i) - T(n) + \sum_i (\sum_k (F_{n,i})) - \sum_i (F_{n,i}) - \sum_i (F_{i,n})$$

. As a summary we have:



$$\text{true positives} := T_n$$

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

$$\text{false positives} := \sum_i (F_{i,n})$$

$$\text{false negatives} := \sum_i (F_{n,i})$$

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

$$\text{sensitivity}(n) := \frac{T_n}{T_n + \sum_i (F_{n,i})}$$

$$\text{specificity}(n)$$

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

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

## Introducing Confidence

We define a confidence<sup>T</sup> confidence<sup>T</sup> which is a measure of how confident we can be that the reply of our classification function is actually correct.

$T_n + \sum_i (F_{i,n})$   $T_n + \sum_i (F_{i,n})$  are all cases where the classification function replied with  $n$  but only  $T_n$  of those are correct. We thus define

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

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

But can we also define a

confidence<sup>⊥</sup> confidence<sup>⊥</sup> 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

$$\sum_i (\sum_k (F_{i,k})) - \sum_i (F_{i,n}) + \sum_i (T_i) - T_n \quad \sum_i (\sum_k (F_{i,k})) - \sum_i (F_{i,n}) + \sum_i (T_i) - T_n$$


all of which are correct except  $\sum_i (F_{n,i})$   $\sum_i (F_{n,i})$ . Thus, we define

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

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

share cite improve this answer edited Sep 27 '18 at 21:07

answered Nov 9 '17 at 12:55

 mroman 266 1 10

Can you please provide any example of calculating Mean Accuracy using confusion matrix. – Aadnan Farooq A Sep 27 '18 at 1:17



You can find a more detailed description with examples here: [mroman.ch/guides/sensspec.html](http://mroman.ch/guides/sensspec.html) – mroman Sep 27 '18 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 '18 at 16:14

[add a comment](#)

## Imbalanced classes in your dataset

5

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

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

He will be right 99% of the time. And therefore gets a 99% accuracy. Can I may 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](#) answered Nov 9 '17 at 17:34



Mayou36

688 3 15

[add a comment](#)

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

2

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](#) answered Sep 27 '18 at 14:27



jeza

407 3 19

[add a comment](#)

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

1

You own an egg shop and each egg you sell generates a net revenue of 22 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 11 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 22 dollars" (positive) and "wants to buy only at 11 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 11 dollar because the customer leaves and you didn't try to make a successful discount
- for every false negative you loose 11 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 22, 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 answered Nov 9 '17 at 17:40



Benoit Sanchez  
5,497 10 30

add a comment



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 answered Nov 9 '17 at 11:05



James

1,135

11

18

- 
- 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 '17 at 11:28

---

(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 '17 at 11:45

---

[add a comment](#)