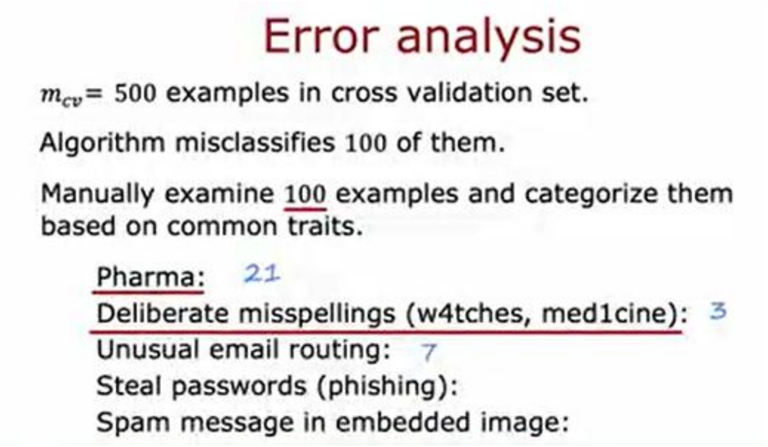
Error analysis

In terms of the most important ways to help you run diagnostics to choose what to try next to improve your learning algorithm performance, I would say bias and variance is probably the most important idea and error analysis would probably be second on my list. Let's take a look at what this means. Concretely, let's say you have m_cv equals 500 cross validation examples and your algorithm misclassifies 100 of these 500 cross validation examples.

The error analysis process just refers to manually looking through these 100 examples and trying to gain insights into where the algorithm is going wrong. Specifically, what I will often do is find a set of examples that the algorithm has misclassified examples from the cross validation set and try to group them into common teams or common properties or common traits.

For example, if you notice that quite a lot of the misclassified spam emails are pharmaceutical sales, trying to sell medicines or drugs then I will actually go through these examples and count up by hand how many emails that are misclassified are pharmaceutical spam and say there are 21 emails that are pharmaceutical spam. Or if you suspect that deliberate misspellings may be tripping over your spam classifier then I will also go through and just count up how many of these examples that it misclassified had a deliberate misspelling.



Error analysis

m_{cv}= 500 examples in cross validation set.

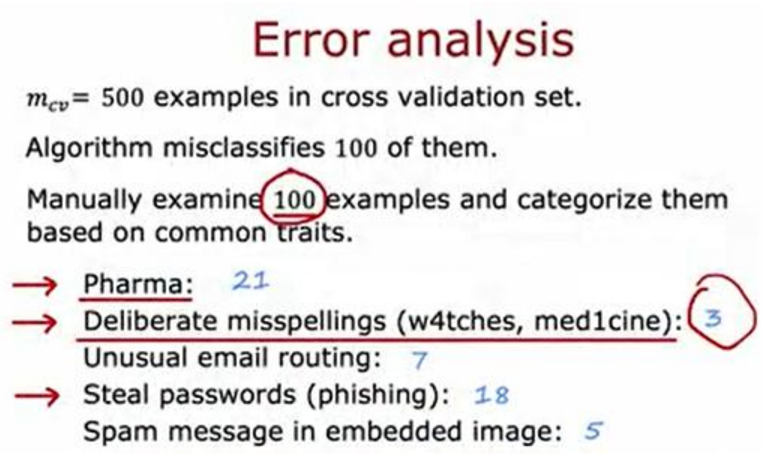
Algorithm misclassifies 100 of them.

Manually examine 100 examples and categorize them based on common traits.

Pharma: 21
Deliberate misspellings (w4tches, med1cine): 3
Unusual email routing: 7
Steal passwords (phishing):
Spam message in embedded image:

Let's say I find three out of a 100. Or looking through the email routing info I find seven has unusual email routing and 18 emails trying to steal passwords or phishing emails. Spam is sometimes also, instead of writing the spam message in the email body they instead create an image and then writes to spam the message inside an image that appears in the email. This makes it a little bit harder for learning algorithm to figure out what's going on.

Maybe some of those emails are these embedded image spam. If you end up with these counts then that tells you that pharmaceutical spam and emails trying to steal passwords or phishing emails seem to be huge problems considering deliberate misspellings, well, it is a problem it is a smaller one. In particular, what this analysis tells you is that even if you were to build really sophisticated algorithms to find deliberate misspellings it will only solve three out of 100 of your misclassified examples.



Error analysis

m_{cv} = 500 examples in cross validation set.

Algorithm misclassifies 100 of them.

Manually examine 100 examples and categorize them based on common traits.

- Pharma: 21
- Deliberate misspellings (w4tches, med1cine): 3
- Unusual email routing: 7
- Steal passwords (phishing): 18
- Spam message in embedded image: 5

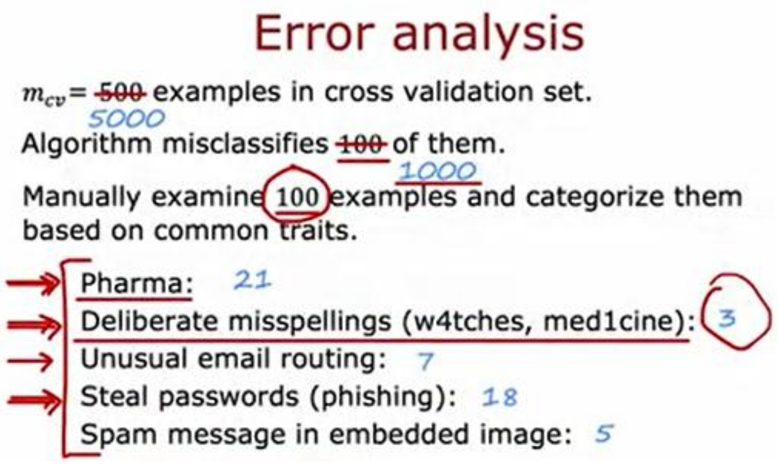
The net impact seems like it may not be that large. Doesn't mean it's not worth doing? But when you're prioritizing what to do, you might therefore decide not to prioritize this as highly. By the way, I'm telling the story because I once actually spent a lot of time building algorithms to find deliberate misspellings and spam emails only much later to realize that the net impact was actually quite small.

This is one example where I wish I'd done more careful error analysis before spending a lot of time myself trying to find these deliberate misspellings. Just a couple of notes on this process. These categories can be overlapping or in other words they're not mutually exclusive.

For example, there can be a pharmaceutical spam email that also has unusual routing or a password that has deliberate misspellings and is also trying to carry out the phishing attack. One email can be counted in multiple categories. In this example, I had said that the algorithm misclassified as 100 examples and we'll look at all 100 examples manually.

If you have a larger cross validation set, say we had 5,000 cross validation examples and if the algorithm misclassified say 1,000 of them then you may not have the time depending on the team size and how much time you have to work on this project. You may not have the time to manually look at all 1,000 examples that the algorithm misclassifies.

In that case, I will often sample randomly a subset of usually around a 100, maybe a couple 100 examples because that's the amount that you can look through in a reasonable amount of time. Hopefully looking through maybe around a 100 examples will give you enough statistics about whether the most common types of errors and therefore where maybe most fruitful to focus your attention.



Error analysis

m_{cv} = ~~500~~ 5000 examples in cross validation set.

Algorithm misclassifies ~~100~~ 1000 of them.

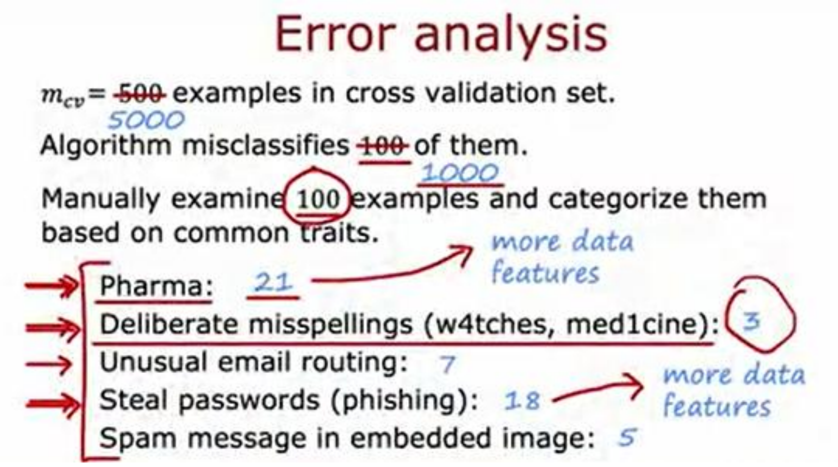
Manually examine 100 examples and categorize them based on common traits.

- Pharma: 21
- Deliberate misspellings (w4tches, med1cine): 3
- Unusual email routing: 7
- Steal passwords (phishing): 18
- Spam message in embedded image: 5

After this analysis, if you find that a lot of errors are pharmaceutical spam emails then this might give you some ideas or inspiration for things to do next. For example, you may decide to collect more data but not more data of everything, but just try to find more data of pharmaceutical spam emails so that the learning algorithm can do a better job recognizing these pharmaceutical spam.

Or you may decide to come up with some new features that are related to say specific names of drugs or specific names of pharmaceutical products of the spammers are trying to sell in order to help your learning algorithm become better at recognizing this type of pharma spam. Then again this might inspire you to make specific changes to the algorithm relating to detecting phishing emails.

For example, you might look at the URLs in the email and write special code to come with extra features to see if it's linking to suspicious URLs. Or again, you might decide to get more data of phishing emails specifically in order to help your learning algorithm do a better job of recognizing them. The point of this error analysis is by manually examining a set of examples that your algorithm is misclassifying or mislabeling.



Error analysis

m_{cv} = ~~500~~ examples in cross validation set.
5000

Algorithm misclassifies ~~100~~ of them.
1000

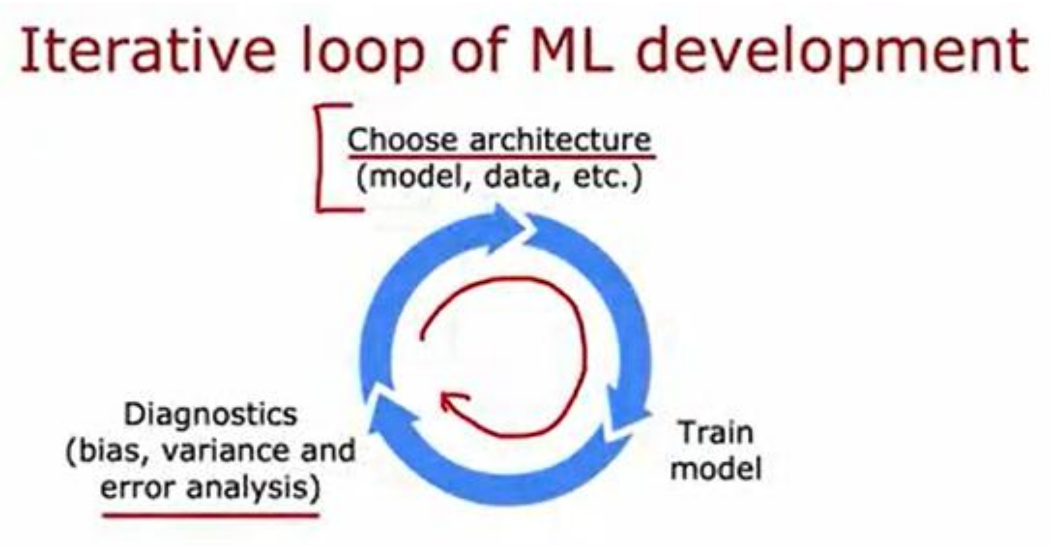
Manually examine 100 examples and categorize them based on common traits.

- Pharma: 21 → more data features
- Deliberate misspellings (w4tches, med1cine): 3
- Unusual email routing: 7
- Steal passwords (phishing): 18 → more data features
- Spam message in embedded image: 5

Often this will create inspiration for what might be useful to try next and sometimes it can also tell you that certain types of errors are sufficiently rare that they aren't worth as much of your time to try to fix. Returning to this list, a bias variance analysis should tell you if collecting more data is helpful or not.

Based on our error analysis in the example we just went through, it looks like more sophisticated email features could help but only a bit whereas more sophisticated features to detect pharma spam or phishing emails could help a lot. These detecting misspellings would not help nearly as much.

In general I found both the bias variance diagnosis as well as carrying out this form of error analysis to be really helpful to screening or to deciding which changes to the model are more promising to try on next. Now one limitation of error analysis is that it's much easier to do for problems that humans are good at. You can look at the email and say you think is a spam email, why did the algorithm get it wrong?



Iterative loop of ML development

Choose architecture
(model, data, etc.)

Train
model

Diagnostics
(bias, variance and
error analysis)

Error analysis can be a bit harder for tasks that even humans aren't good at. For example, if you're trying to predict what ads someone will click on on the website. Well, I can't predict what someone will click on. Error analysis there actually tends to be more difficult. But when you apply error analysis to problems that you can it can be extremely helpful for focusing attention on the more promising things to try. That in turn can easily save you months of otherwise fruitless work. In the next video, I'd like to dive deeper into the problem of adding data.

When you train a learning algorithm, sometimes you decide there's high variance and you want to get more data for it. Some techniques they can make how you add data much more efficient. Let's take a look at that so that hopefully you'll be armed with some good ways to get more data for your learning application.