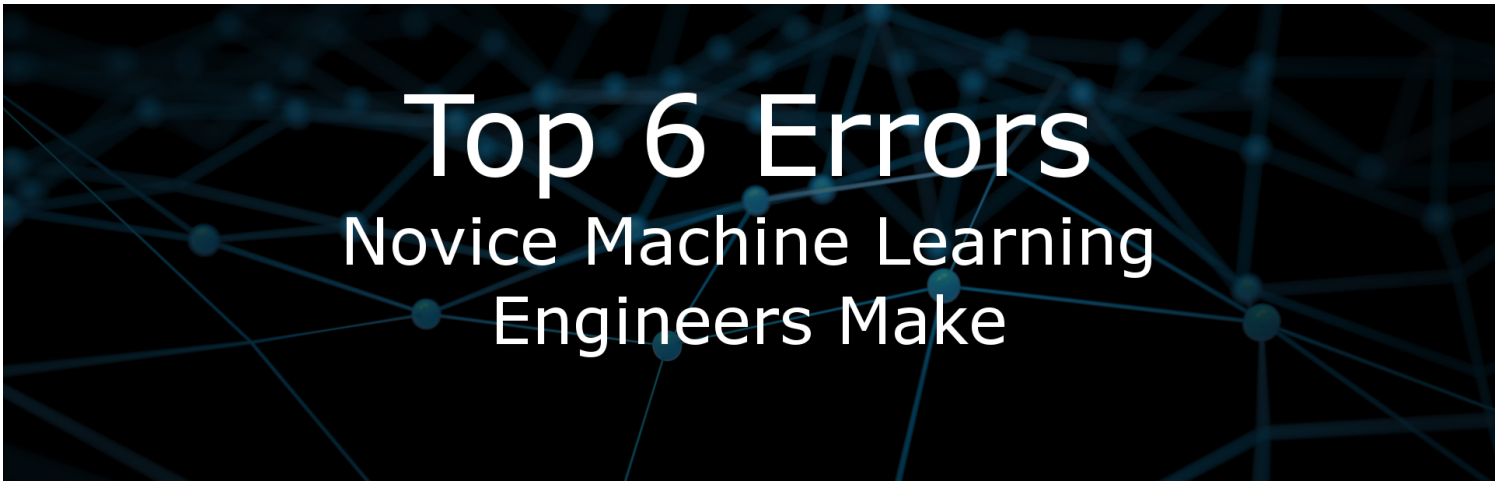
Christopher Dossman Follow
Machine Learning Engineer, Teacher, and Entrepreneur
Oct 15 · 5 min read



Top 6 errors novice machine learning engineers make

In machine learning, there are many ways to build a product or solution and each way assumes something different. Many times, it's not obvious how to navigate and identify which assumptions are reasonable. People new to machine learning make mistakes, which in hindsight will often feel silly. I've created a list of the top mistakes that novice machine learning engineers make. Hopefully, you can learn from these common errors and create more robust solutions that bring real value.

Taking the default loss function for granted

Mean squared error is great! It really is an amazing default to start off with, but when it comes to real-world applications this off-the-shelf loss function is rarely optimum for the business problem you're trying to solve for.

Take for example fraud detection. In order to align with business objectives what you really want is to penalize false negatives in proportion to the dollar amount lost due to fraud. Using mean squared

error might give you OK results but will never give you state of the art results.

Take Away: Always build a custom loss function that closely matches your solution objectives

Using one algorithm/method for all problems

Many will finish their first tutorial and immediately start using the same algorithm that they learned on every use case they can imagine. It's familiar and they figure it will work just as well as any other algorithm. This is a bad assumption and will lead to poor results.

Let your data choose your model for you. Once you have preprocessed your data, feed it into many different models and see what the results are. You will have a good idea of what models work best and what models don't work so well.

Becoming a Machine Learning Engineer | Step 2: Pick a process
Check out this article and get a handle on your process.

Take Away: If you find yourself using the same algorithm over and over again it probably means you're not getting the best results.

Forget about outliers

Outliers can be important or completely ignored, just based on context. Take for example revenue forecasting. Large spikes in revenue can occur and it is a good idea to look at them and understand why they occurred. In the case of outliers caused by some type of error, it is safe to ignore them and remove from your data.

From a model perspective, some are more sensitive to outliers than other. Take for example Adaboost, it treats those outliers as "hard" cases and puts tremendous weights on outliers while decision trees might simple count each outlier as one false classification.

Becoming a Machine Learning Engineer | Step 2: Pick a process
Goes over best practices that you can use to avoid this mistake

Take Away: Always look at your data closely before you start your work and determine if outliers should be ignored or looked at more closely

Not properly dealing with cyclical features

Hours of the day, days of the week, months in a year, and wind direction are all examples of features that are cyclical. Many new machine learning engineers don't think to convert these features into a representation that can preserve information such as hour 23 and hour 0 being close to each other and not far.

Keeping with the hour example, the best way to handle this is to calculate the sin and cos component so that you represent your cyclical feature as (x,y) coordinates of a circle. In this representation hour, 23 and hour 0 are right next to each other numerically, just as they should be.

Take Away: If you have cyclical features and you are not converting them you are giving your model garbage data to start with.

L1/L2 Regularization without standardization

L1 and L2 regularization penalizes large coefficients and is a common way to regularize linear or logistic regression; however, many machine learning engineers are not aware that is important to standardize features before applying regularization.

Assuming you had a linear regression model with a transaction amount as a feature. Without regularization, if the transaction amount is in dollars, the fitted coefficients are going to be around 100 times larger than if they were in cents. This will cause a bias and tend to penalize features that are smaller in scale. To avoid the problem, standardize all the features and put them on equal footing so regularization is the same all over your features.

Take Away: Regularization is great but can cause headaches if you don't have standardized features

Interpreting absolute value of coefficients from linear or logistic regression as feature importance

Many off-the-shelf linear regressors return p-values for each coefficient, many novice machine learning engineers believe that for linear models, the bigger the value of the coefficient, the more important the feature is. This is hardly ever true as the scale of the variable changes the absolute value of the coefficient. If the features are co-linear, coefficients can shift

from one feature to the other. The more features the data set has the more likely the features are co-linear and the less reliable simple interpretations of feature importance are.

Take Away: Understanding what features are most important to a result is important, but don't assume that you can look at the coefficients. They often don't tell the whole story.

. . .

Doing a few projects and getting good results can feel like winning a million bucks. You worked hard and you have the results to prove that you did a good job, but just like with any other industry the devil is in the details and even fancy plots can hide bias and error. This list is not meant to be exhaustive, but merely to cause the reader to think about all the small issues that might be hiding in your solution. In order to achieve good results, it is important to follow your process and always double check that you are not making some common mistakes.

If you found this article useful you will get a lot out of my **Becoming a machine learning engineer | Step 2: Picking a Process** article. It helps you iron out a process that will allow you to catch more simple mistakes and get better results.

Thanks for reading :) If you enjoyed it, hit that clap button below as many times as possible! It would mean a lot to me and encourage me to write more stories like this

Let's also connect on Twitter, LinkedIn, or email