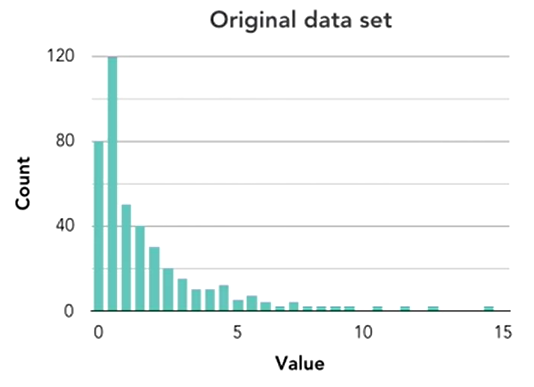
*Start a new Python project folder … do not reuse the previous workspace … As with any keyboard-driven console-like environment, developing muscle -memory for the common commands is also part of the learning curve.*

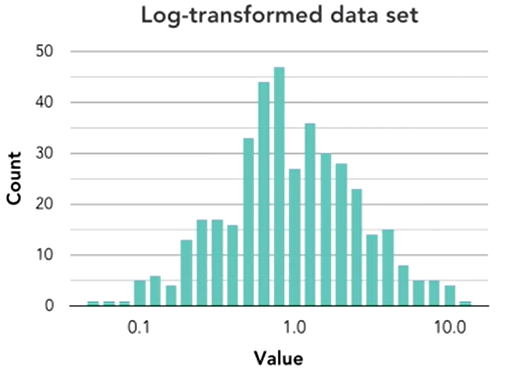
***Feature engineering*** is the process of creating new features or transforming your existing features to get the most out of your data. This process of preparing data for a model build includes reading in messy raw text, cleaning data by removing punctuations, removing stop words, stemming, lemmatizing, and vectorizing the data.

In the context of determining if a text message is a spam or ham, what else could we extract from the text that would be helpful for the model to decipher spam from ham? For instance, maybe we could include the length of the text field. Maybe spam tends to be a little bit longer than the real text messages. Or maybe we could include what percent of the characters in the text message are punctuation. Maybe real text messages underuse punctuation. Or maybe what percent of characters are capitalized are indicative of whether it’s spam or not. So that’s a couple of ideas of some “features” that you could create that that would help our model indentify spam from nonspam. A new feature is added to the matrix as a new column.

So given these new features, or really any other already existing features, maybe you need to apply some sort of transformation to your data to make it more well-behaved. One broad popular type of transformations are called “power transformations.” So this would include squaring your data, taking the square root, et cetera. One example of why you might need to transform your data would be if you have a very skewed data set with a very long right tail where you have a lot of outliers.



In that case, you might want to apply a log transformation which basically pulls that long tail and all those outliers back towards the bulk of the data.



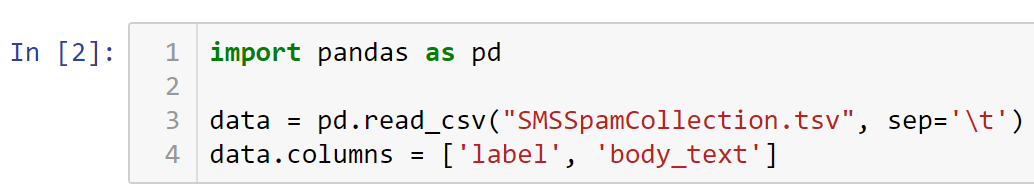
And what this does is it helps the model draw correlations and better understand the data without trying to overfit to that long tail and those outliers.

Another topic that falls under transformations is standardizing your data, or transforming it all to be on the same scale. Some models perform better when all features are on the same scale. It’s especially important as you get to this phase that you’re keeping the problem context in mind.

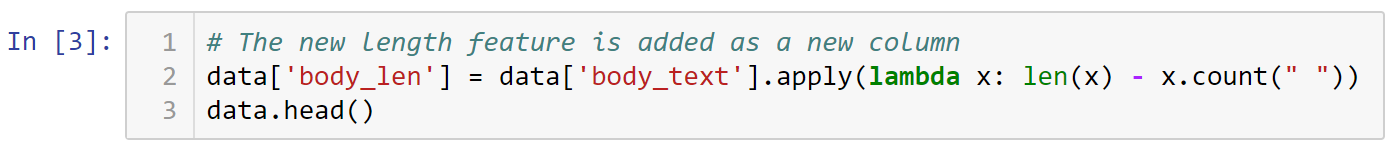
When you do feature creation you’re always trying to imagine what additional information might be helpful to the model within the context and understanding of what exactly it’s trying to predict. For instance, as an extreme example in our spam detection problem, the number of ‘A’s that appear in a text message likely isn’t predictive of whether this is spam or ham. But maybe the amount of punctuation or the length of a text message is. So always keep your problem in mind. And this is the stage where you’re allowed to get a little bit creative to try to extract as much value out of your data as possible.

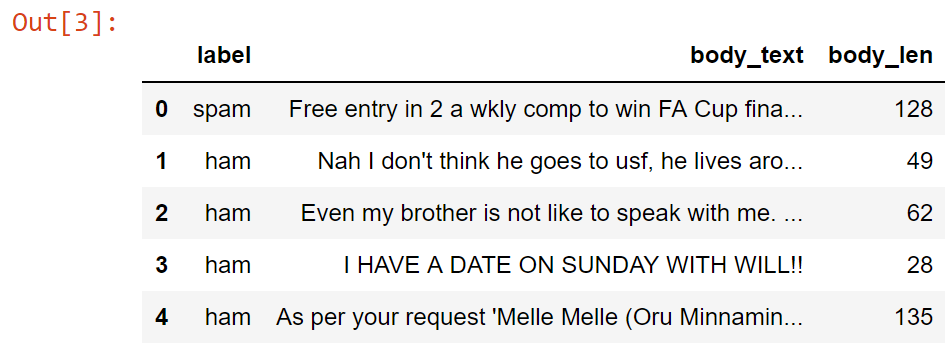
**Feature Creation**

Again, we do this to try to extract as many values out of the given data as we can by imagining what additional information would be helpful for the model to make accurate predictions. We are only going to touch on very high-level concepts just so we can get a basic working prototype. In creating feature(s) it is advisable to start with the raw text and not the clean text. This principle will be explained shortly.

**Creating feature for text message length**

The first feature we are going to create is a text message length feature. We are going to work under the hypothesis that spam messages tend to be longer than real text messages. So we will create this feature and then we will explore whether our hypothesis is accurate. To have an accurate character count, white spaces are discarded using the expression, ‘- x.count(“ “)’.

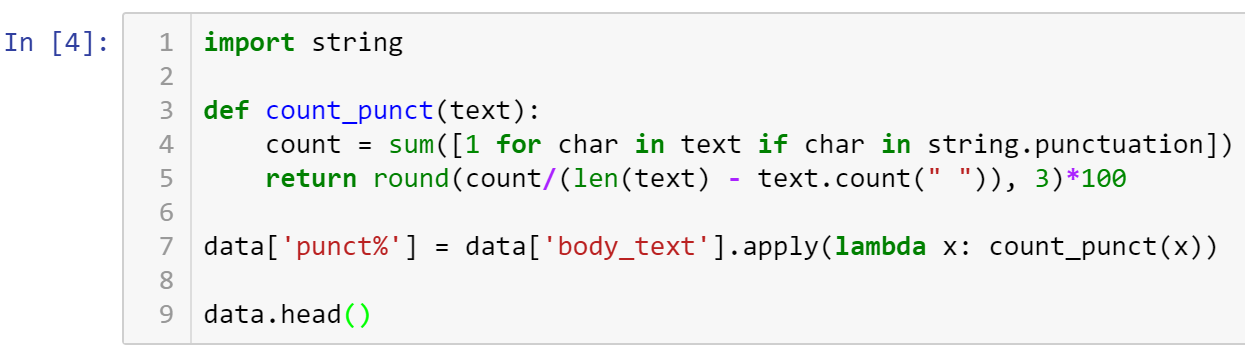




**Creating feature for % of text that is punctuation**

For this next feature, we are going to generate the percent of each text message that is punctuation. Again, we are going to work under the hypothesis that real text messages use less punctuation than spam. This is one of the reason why we use the raw text and not the clean text because the clean text has no punctuation.

*Note in the function below, the first character after the parenthesis is the number one, not the letter I. This means that the operation will return a one if it finds a punctuation. Then what the function will do is sum the number of ones being returned.*



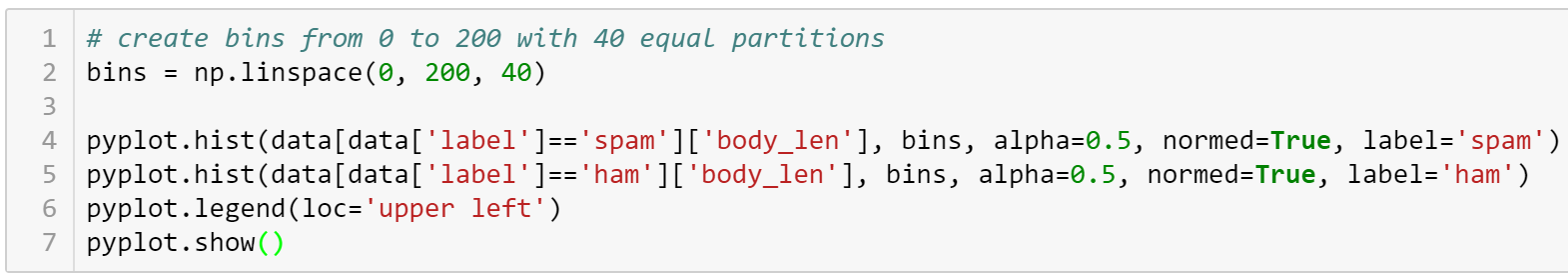


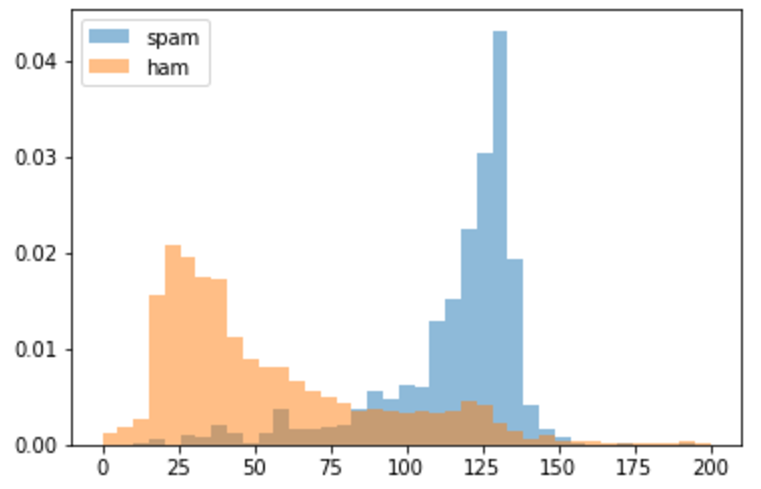
**Evaluating what these new features look like**

Again, the goal here is to generate new features (indicated as new columns) that help a model distinguish spam from real text messages. So it is always useful to find some way to see if your new features appear to be predictive, or correlated to the response in some way. One way of evaluation is to use overlayed histograms to look at the value of these created features.

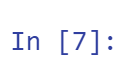


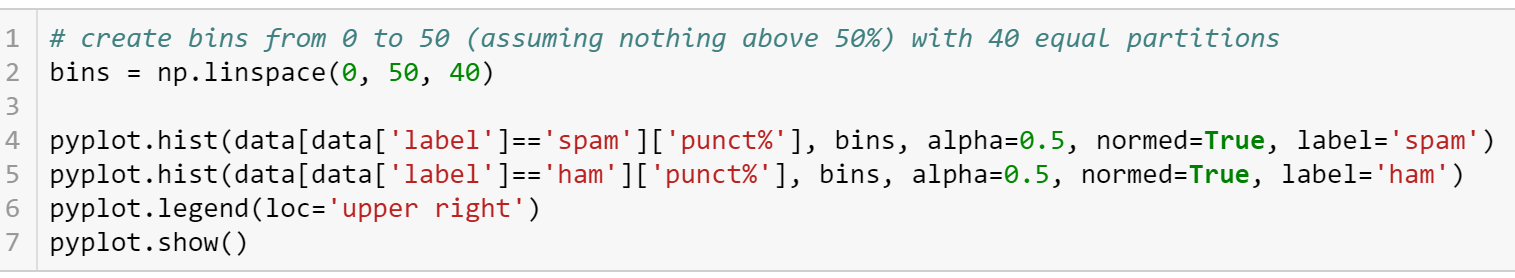


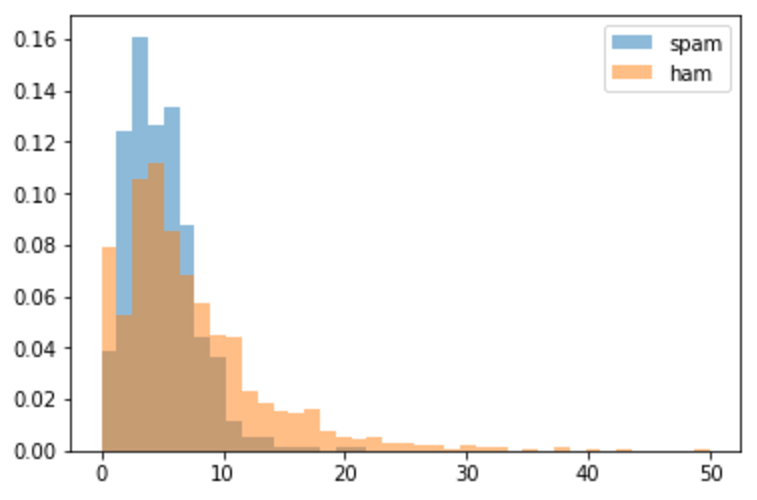




From the graph above, you can see that the body length is very different for ham versus spam. So spam text messages seem to be quite a bit longer than regular text messages. It appears that this extra feature could be really helpful for the model to distinguish ham from spam. So, if we didn’t create this feature, the model hay not necessarily pick up on this difference. Now, graphing punctuations ...







Interpretation …. Based on the graph above, you can see that there is not nearly as much of a difference in punctuation use. You can see that spam might be a little bit more concentrated here on the left, whereas ham tends to have more of a tail over the right-hand side. However, it is pretty clear which one of these new features is likely to help out the model the most.

So in terms of our original hypotheses, our hypothesis that spam messages tend to be longer than non-spam messages seems to be correct based on this evaluation, and this feature is likely to provide some value to the model. However, our hypothesis that ham messages contain less punctuation than spam doesn’t appear to be accurate, and it isn’t quite clear whether this feature will provide value to the model. Now, in cases like this where there is some separation between the distributions, typically we will err on the side of leaving this feature in the model just to see what kind of value the model itself may be able to extract out of it. This exercise is an example of how you might evaluate whether some newly created features will be useful to the model.

● All submissions should be separate from other exercises and quests. Please do not lump all your answers into one document and re-using that same workspace to gain multiple points. Thanks.

● Place your name at the bottom of your code, download your Python program in html format, and submit your work in Canvas.