(https://google.com/racialequity?authuser=0)

Classification: Prediction Bias

ated Time: 7 minutes

Logistic regression predictions should be unbiased. That is:

"average of predictions" should ≈ "average of observations"

Prediction bias is a quantity that measures how far apart those two averages are. That is:

$$\text{prediction bias} = \text{average of predictions} - \text{average of labels in data set}$$

'Prediction bias" is a different quantity than bias ://developers.google.com/machine-learning/crash-course/descending-into-ml?authuser=0) (the *b* in wx +

A significant nonzero prediction bias tells you there is a bug somewhere in your model, as it indicates that the model is wrong about how frequently positive labels occur.

For example, let's say we know that on average, 1% of all emails are spam. If we don't know anything at all about a given email, we should predict that it's 1% likely to be spam. Similarly, a good spam model should predict on average that emails are 1% likely to be spam. (In other words, if we average the predicted likelihoods of each individual email being spam, the result should be 1%.) If instead, the model's average prediction is 20% likelihood of being spam, we can conclude that it exhibits prediction bias.

Possible root causes of prediction bias are:

- Incomplete feature set
- Noisy data set
- Buggy pipeline
- Biased training sample
- Overly strong regularization

You might be tempted to correct prediction bias by post-processing the learned model—that is, by adding a **calibration layer** that adjusts your model's output to reduce the prediction bias. For example, if your model has +3% bias, you could add a calibration layer that lowers the mean prediction by 3%. However, adding a calibration layer is a bad idea for the following reasons:

- You're fixing the symptom rather than the cause.
- You've built a more brittle system that you must now keep up to date.

If possible, avoid calibration layers. Projects that use calibration layers tend to become reliant on them—using calibration layers to fix all their model's sins. Ultimately, maintaining the calibration layers can become a nightmare.

A good model will usually have near-zero bias. That said, a low prediction bias does not prove that your m
d. A really terrible model could have a zero prediction bias. For example, a model that just predicts the me
or all examples would be a bad model, despite having zero bias.

Bucketing and Prediction Bias

Logistic regression predicts a value *between* 0 and 1. However, all labeled examples are either exactly 0 (meaning, for example, "not spam") or exactly 1 (meaning, for example, "spam"). Therefore, when examining prediction bias, you cannot accurately determine the prediction bias based on only one example; you must examine the prediction bias on a "bucket" of examples. That is, prediction bias for logistic regression only makes sense when grouping enough examples together to be able to compare a predicted value (for example, 0.392) to observed values (for example, 0.394).

You can form buckets in the following ways:

- Linearly breaking up the target predictions.
- Forming quantiles.

Consider the following calibration plot from a particular model. Each dot represents a bucket of 1,000 values. The axes have the following meanings:

- The x-axis represents the average of values the model predicted for that bucket.
- The y-axis represents the actual average of values in the data set for that bucket.

Both axes are logarithmic scales.

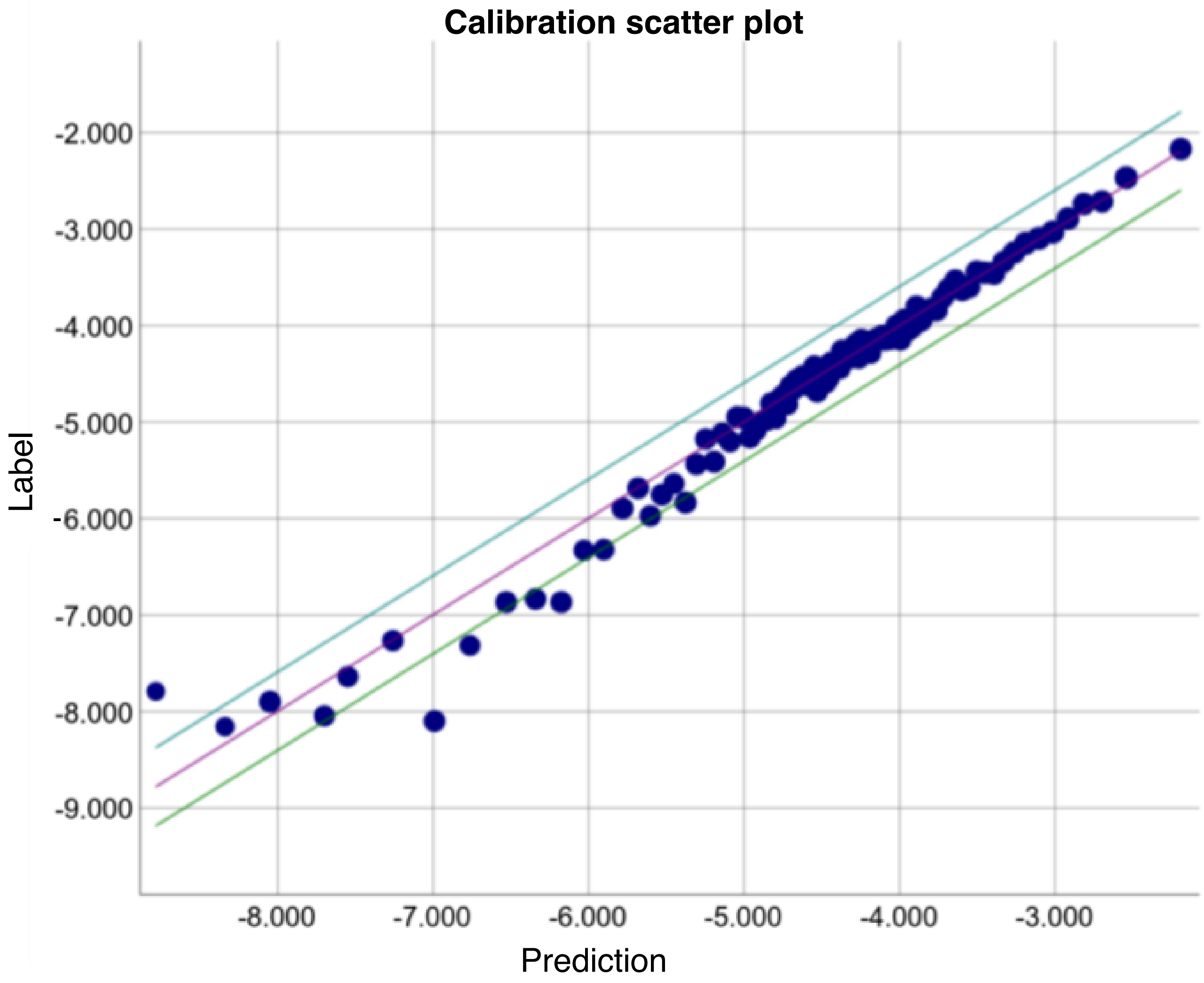

Figure 8. Prediction bias curve (logarithmic scales)

Why are the predictions so poor for only *part* of the model? Here are a few possibilities:

- The training set doesn't adequately represent certain subsets of the data space.
- Some subsets of the data set are noisier than others.
- The model is overly regularized (https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/video-lecture?authuser=0). (Consider reducing the value of lambda (https://developers.google.com/machine-learning/glossary?authuser=0#lambda).)

ɜrms

cketing ://developers.google.com/machine-ıg/glossary?authuser=0#bucketing)

ɜdiction bias

- calibration layer (https://developers.google.com/machine-learning/glossary?authuser=0#calibration_layer)