“Random State” Module Transcript

Chapter 1

Introduction to the Module

Hi, I'm Hayley, and I'm excited to welcome you back to the Global Settings Module Series. In this mini-module, we will talk about random state settings, which are particularly important for ensuring reproducibility of model results.

Chapter 2

Random State Overview

In machine learning, the random state parameter initializes the random number generator for tasks involving randomness, such as splitting datasets into train and test sets, initializing model parameters, or conducting random sampling.

In One AI, the random state is a pseudo-random number parameter that allows you to reproduce the same train-test split each time the model is run. The default value is 43, and if unchanged, the model will use consistent train-test splits. To obtain a different split, change this value to any number between 0 and 4,294,967,295. I know that's a crazy range, but don't blame me. We did not invent this concept.

It's really important to understand that the random state should not be treated as a hyperparameter related to your model's performance. It is not a scale of randomness, but instead a seed value that ensures random number generators consistently produce the same results.

We recommend leaving this setting unchanged in most One AI machine learning use cases for a few reasons. First, setting a specific random state ensures the random number generator produces the same sequence of random numbers each time the code is executed, allowing for reproducible results. This is crucial for debugging, testing, and comparing different machine learning models or parameters.

Next, when splitting a dataset into training and test sets, specifying a random state ensures that the same data points are assigned to the training and test sets consistently. This guarantees consistency in model evaluation and validation.

And finally, using a random state when shuffling datasets ensures the same shuffled order is produced each time. This consistency is important for maintaining uniformity in data steps and ensuring models are trained on the same data distribution across different runs. However, changing the random state between runs can be beneficial in certain edge cases.

First, it allows you to obtain multiple performance estimates and assess the variability in model performance due to randomness in the data splitting. Next changing the random state can help diagnose issues related to randomness in the code or data processing pipeline. And finally, it is useful for assessing the stability of machine learning models by observing how performance changes with different data splits.

Chapter 3

Adjusting Random State in One Model

Adjust the random state setting by toggling the override to on and entering a new value within the random state range in the designated field. For example, we can input "500" like so. Remember to scroll to the bottom and save your changes before rerunning your model.

Chapter 4

Conclusion & Next Steps

Thanks for joining me to take a look at the random state setting, and congratulations on completing the Global Settings Module Series! Check out our Advanced Configuration Series where we cover topics like dimensionality reduction, estimator configuration, upsampling, and per column interventions.

These modules will allow you to continue learning how to adjust your model configurations for full control over how your model runs and to enhance performance.

Happy modeling!