

# **“Leakage & Suspicious Performance Threshold” 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 explore leakage and suspicious performance threshold configurations. These topics are grouped together because they are both measured by the same test. The suspicious performance threshold is simply a less stringent version of the leakage performance threshold.

## **Chapter 2**

### **Leakage Performance Threshold Overview**

As we briefly discussed in the Correlation Type module, each time a user runs a machine learning model in One AI, a data leakage check is performed during the data preprocessing before any machine learning occurs.

Data leakage occurs when the training dataset contains information about the target variable that won't be available when making predictions on new, unseen data. In simpler terms, it's a "cheat" column that predicts the outcome too well to be realistic.

For example, a flag in the data indicating whether someone is a future termination would be considered data leakage when the model is predicting attrition. It's okay to have this data in One Model, but it shouldn't be included as an input variable.

Similar to a student, a model isn't truly learning if you give it the answer upfront. We want the model to make reliable predictions without cheat columns and learn from valuable features that provide rich insights into what actually drives the prediction.

Leaving leaking columns in the training data can inflate performance metrics, giving the false impression of a stronger model, but the model ultimately shouldn't be trusted or used on unseen data until the leakage is removed. In One AI, leakage is identified by generating a random forest model against the target using only the feature in question and measuring the performance with an ROC-AUC score. By default, a score of 0.85 is considered leakage, and the variable will be automatically dropped.

## **Chapter 3**

### **Adjusting the Leakage Performance Threshold in One AI**

This threshold can be adjusted by toggling the Override slider to 'On' and entering a new value ranging from 0 to 1 in the designated field. For example, entering 0.9 would require a slightly higher ROC-AUC score in order for leaking columns to be automatically dropped by One AI.

You should adjust this setting if the model is incorrectly dropping columns that are not cheat columns or if it's missing actual cheat columns. However, this is quite rare. You can also perform a per column intervention to prevent specific columns from being dropped while keeping the data leakage threshold in place for all other columns.

If you configure this threshold, remember to scroll to the bottom of the screen and save before rerunning your model.

## **Chapter 4**

### **Suspicious Performance Threshold Overview**

As I said in the introduction, the suspicious performance threshold is a less stringent version of the data leakage performance threshold. It uses the same detection test to inform users of possible leakage. The default suspicious performance threshold is 0.7 compared to the 0.85 for leakage.

Suspicious variables are not automatically dropped like those for leakage, but are flagged with a purple label in the EDA report. These variables should be validated to ensure they do not allow the model to cheat. If you determine that there is leakage present, you should edit your model to exclude the column or columns in the core attribute section of the One AI Query Builder. If leakage is not present, no action is necessary.

## **Chapter 5**

### **Adjusting the Suspicious Performance Threshold in One AI**

This threshold can be adjusted in exactly the same way as the leakage performance threshold setting - by toggling the Override slider to 'On' and entering a new value ranging from zero to one in the designated field. For example, you could input 0.75. Don't forget to save your threshold changes before rerunning your model.

## **Chapter 6**

### **Conclusion & What's Next**

Thanks for joining me to discuss leakage and suspicious performance threshold settings. In the next mini-module, we will examine the category size threshold. Happy modeling!