“Per Column Interventions” Module Transcript

Chapter 1

Intro, Topics Covered, & Learning Outcomes

Howdy, folks. My name is Austin, and I'm a machine learning engineer on the One AI team here at One Model. In the "Data Preprocessing" module, you learned how the raw data within your model dataset is transformed into a format that is suitable for training machine learning models. Data preprocessing settings apply to the entire model dataset, but sometimes individual columns have distinct characteristics. So applying the same preprocessing to all features is not ideal. In this module, we will discuss per column interventions, which allow users to make changes specifically for individual features in the dataset.

We will cover an overview of per column interventions in machine learning, the types of per column interventions available to use with One AI models and when they are most appropriate to utilize, and how to configure each type of per column intervention in One Model.

After completing this module, you will understand what per column interventions are; and when and how to apply different per column interventions, ensuring that each feature is optimally processed. You will understand how to configure column droppability settings and select appropriate null-filling strategies to maximize the use of available data and improve model accuracy. And finally, you will learn how to apply tailored interventions for categorical and continuous features, ensuring that each type of data is handled effectively to enhance the overall model performance and interpretability.

Chapter 2

Per Column Intervention Overview

In machine learning, per column interventions refer to the application of specific preprocessing and transformations to individual columns or features in the model dataset, rather than applying the same processing to the entire dataset. This approach acknowledges that different columns have unique characteristics and requirements, calling for tailored preprocessing. Interventions on a per column basis involve actions such as cleaning, transforming, handling missing values, normalizing or scaling data, encoding categorical variables, and other feature engineering techniques.

These techniques are applied based on each column's specific properties and the needs of the machine learning model. When done effectively, per column interventions can improve model performance and interpretability.

By tailoring preprocessing to the specific needs of each feature, models can better capture the underlying patterns and relationships in the data, leading to a more accurate prediction. When each feature is processed appropriately, it becomes easier for the model to learn the true importance of each feature, enhancing the overall model accuracy and reliability.

They also help reduce the risk of overfitting. Applying suitable transformation to individual features can reduce noise and potential biases in the data, resulting in cleaner and more reliable inputs from the model. For instance, handling outliers and numerical data through techniques like clipping or scaling can prevent the model from learning misleading patterns. Additionally, they assist in handling diverse data types within a single dataset by ensuring that each feature is appropriately processed rather than using a single technique that may be suitable for most features, but not ideal for each one.

Finally, they allow users to choose how missing values should be handled depending on the feature type. This prevents important columns from being dropped due to excessive null data. This approach maximizes the available data. Instead of discarding entire rows or columns with missing values, null-filling techniques can recover valuable information.

Chapter 3

Per Column Interventions Available in One AI

Let's talk about some of the per column intervention options available in One AI. One AI offers several different types of per column interventions to allow for customized preprocessing of individual features, ensuring that each one is handled in the most effective way for the specific dataset and model requirements. I will go through each of them now:

You have the option to modify the droppability of each column. There are three droppability options. Number one, 'Is Droppable', is the default setting for all columns. This column can be selected if it is predictive and improves model accuracy, or it can be dropped by One AI if it doesn't enhance performance or if it violates a global setting. It will go through the standard feature selection process. Number two, 'Not Droppable', means that the column cannot be dropped from the model by One AI. While this doesn't guarantee that the column will be selected by the model, it will always be processed and

at least considered even if it violates global settings. For example, if a column has more null values than your null drop threshold, but it is predictive, it will be selected regardless. This is useful for trying individual columns that don't meet global settings without altering the settings for the entire model. Lastly, we have 'Always', which ensures that the column will always be dropped and is not tested in any way. This is more easily done in the Core Attributes section of the One AI Query Builder, but it can be done here if desired.

Next, you have the option to configure null-filling strategies for selected columns. Null values can negatively impact performance, so addressing them is helpful for accurate predictions. A good null fill strategy allows a column to be used in the model dataset despite exceeding the null drop threshold, making more attributes available for the model to try. This can be a good alternative to changing a column's droppability. We have several strategies available for use.

We have the 'Mean' and the 'Median', which are the average or middle values of a continuous column, and those will be used to fill the null values. These cannot be used on categorical columns. Then we have the 'Mode', which is the highest occurring value, and that will be used to fill the null values. These are more straightforward and simple options, but there are other options as well.

B fill and f fill, which are short for backwards fill and forward fill, are particularly useful for dealing with time series data. For b fill, null values will be filled starting from the last value, working towards the first with nulls being filled with the next populated value in the column. For f fill, it's just the opposite. Null values will be filled starting from the first value working towards the last. Any null values will be filled with the previous populated value. Pad is just another term for f fill.

Finally, there's a custom strategy that allows you to use a specific value for all the nulls. The values can be a numerical, date, or categorical value.

The next per column intervention available is the Type Specific Intervention. These interventions are used to customize how individual columns are processed and included in the model, ensuring that each feature is handled in the most appropriate way for optimal model performance.

For categorical features, there are two options. First, 'Exclude', which allows users to enter the names of values to be excluded from being considered as features in the model. Then there is 'Select', which allows you to enter the names of values that the model will then force as selected features. For example, if you have a column called 'Fruit' with apples, bananas, and pears, you can use 'Exclude' to exclude apples or 'Select' to include bananas.

For continuous features, there are also two options. First, there is 'Force Select', which you can use to force this variable to be selected as a feature. Then, there is 'Continuous Strategy', where you can change the continuous strategy for the specific column to either bin or scale. 'Scale' is the default if you didn't make any changes in the Global Settings. But if changed to 'Bin', you can also change the Bin Type here by inputting free text. Check out the "Continuous Strategy" mini-module in the Global Settings series for more information.

Chapter 4

Configuring in One AI

To configure per column interventions in One AI, select the model of interest, then click 'Edit', and scroll down to "Per Column Interventions". Click the Override from 'Off' to 'On, and then click the caret to drop down to the per column interventions selection screen. Click the 'Column Interventions', and then here, you can add a column upon which you would like to perform the per column intervention. Once you've added the column, then you'll click the plus and that will allow you to do the actual per column interventions.

So for example, here you can select your Droppable setting. You can click the Override button for the Null Fill settings And here, this will allow you to pick the strategy, mean, median, mode, etcetera, or you can provide a custom fill value, which you can fill in yourself.

And then we can talk about the Type Specific Interventions. So let's say that your column is categorical. You would select 'Categorical', drop this down. You'll use the override button. And as mentioned, if your column is fruit, then you might want to exclude banana, but you might want to include apples. Similarly, with continuous, if this is a continuous column, then you can use 'Force Select' to force the selection of it. Or you can pick a Continuous Strategy in which you select either 'Bin' or 'Scale'.

If you would like to keep adding interventions, then simply select a new column, hit the plus icon. Now you're ready to do it all over again.

Chapter 5

Conclusion & Thanks

We've explored the importance and implementation of per column interventions in One AI. By understanding and applying these interventions, you can tailor preprocessing steps to the unique characteristics of each feature, improving model accuracy and interpretability. You now have the tools to configure droppability settings, handle missing values effectively, and apply type specific interventions, ensuring that your models are optimized for the best performance. With these skills, you're better equipped to manage your datasets, leading to better models. Happy modeling!