“Generative Attributes” Module Transcript

Chapter 1

Intro, Topics Covered, & Learning Outcomes

Hi, all. My name is Hayley Bresina, and I'm on the One AI team here at One Model. In the previous module, you learned about the process of selecting core attributes to build the datasets used for training models and making predictions. Core attributes are features or columns that already exist in your One Model data. In this module, we are going to talk about using these preexisting features in the dataset to create new generative attributes through some form of transformation, combination, or aggregation.

These generative attributes are created with the intention of providing the machine learning model with additional, more meaningful information that might not be explicitly captured by the original features.

We will cover what generative attributes are, the different types available in One AI, how to create, manage, and utilize generative attributes in your models, and how to validate your generative attributes once they've been created.

Upon completion of this module, you will understand the concept of generative attributes and their importance in providing additional meaningful information beyond the original dataset.

You will know how to create effective generative attributes in One AI by selecting appropriate metrics, time selections, groupings, and filters.

You will understand the importance of validating generative attributes to ensure data integrity and model readiness before running machine learning models, and you will gain insight into how generative attributes can improve model performance by enriching the feature set and enabling the discovery of more complex patterns in the data.

Chapter 2

Generative Attributes Overview

Section 2 - Generative Attributes Overview

Generative attributes are new input variables derived from the original model dataset that are contenders for being selected as features in machine learning models. They are also commonly known as engineered features.These features exist because the way we model data for analytic purposes is often different from how we want data structured for machine learning.

A standard non-generative attribute is something about a person or job or anything else you're making predictions on at a point in time, such as their salary or manager's name. A generative attribute can be a count of events, such as number of promotions over the last five years, a flag if a person meets a set of criteria such as quality hire or higher level aggregations such as team or location, such as team headcount.

As you learned in previous modules, this process is generally part of the broader feature engineering process.Generative attributes allow us to understand things better by creating new ways to look at the information we already have, which can help the machine learning algorithm find helpful patterns that improve predictions.

Therefore, providing models with a wide variety of generative attributes is a great way to increase model performance and robustness because they transform raw data into more meaningful features that can be richer in predictive value than what's available on individual records. Additionally, they help us discover complex patterns and relationships between features within the dataset that may not be obvious without further analysis.

Chapter 3

Types of Generative Attributes

Section 3 - Types of Generative Attributes Available in One AI

There are three different types of generative attributes that we can create in One AI. Starting simple with metrics, generative attributes can be just an existing metric on your One Model site tied to the same time period as your model based query dataset. For example, a generative attribute that is how many times an employee has been promoted in the last year. These are still very helpful because we don't always have columns that measure these types of counts.

Getting slightly more elaborate, let's talk customized metric generative attributes. Metrics from your One Model can be filtered independently and can be tied to periods of time that differ from the model based query dataset. Binary attributes can also be created or users can fill nulls with zeros.

These capabilities can be combined in really cool ways. For example, you could look at if an employee has been promoted with an event reason of progression for all time and make it binary. This attribute would measure if an employee has ever been promoted due to progression in their entire work history at the org or not.

Converting an attribute to binary means that instead of measuring how many times something has happened or to what degree, we are transforming it so that it can only take on two possible values, a 0 meaning it did not happen or a 1 meaning it did. This can be helpful for simplification or if we just want to classify an attribute into two classes.

Filling nulls with zeros is really important because otherwise many generative attributes will be dropped due to missing data, meaning the column is too null for the model to use. In the example I mentioned before, if we did not convert to binary, everyone in the model population who has not been promoted due to progression would have a null value for this generative attribute.

Converting nulls to zeros would ensure everyone has a value and zero is representative of their condition. They have been promoted due to progression zero times. If you convert an attribute to binary, you do not need to also fill with nulls with zeros. In fact, you will not be able to. This is because when we convert to binary, there are no nulls. Everyone gets a value of 0 or 1.

Finally, and maybe the coolest thing is that generative attributes enable aggregations that differ from the model's base query dataset.

Your base query is typically as granular as possible, usually a single employee or event, but attributes about each person don't tell the whole story about what they experience in your organization.

Generative attributes can be created to be grouped by team, plant, office, or any grouping you have an ID for. For example, you could create a team terminations attribute grouped by manager ID, which counts how many terminations have occurred on an employee's team. Other examples include plant safety incidents or office location bonuses.

It can also be beneficial to create multiple versions of a generative attribute that are slightly different, such as converting one to binary or changing up the time period the attribute covers. For example, a binary promotions attribute for the last two years, a promotion attribute for all time, and a team promotions attribute. If they are too similar and correlated, One AI will only keep the one that results in the best performance and fit for your model, and the rest will be automatically dropped.

In the next section, we will hop over to the demo site to talk about how to build each type of generative attribute.

Chapter 4

Creating & Managing Generative Attributes

Section 4 - Creating and Managing Generative Attributes in One AI

Generative attributes are created and managed from the One AI query builder. This is where you create models with recipes or build custom models. To get there, click 'Edit' on the model you are working with and then the 'Configure One AI Recipe' button. Once that loads, scroll down to step 7 - "Which generative attributes do you want to use in your prediction?" and click to expand.

If any users have already created generative attributes on your site, they will appear here. Generative attributes are similar to metrics in Explore in that once any user has created a generative attribute in One Model, all users building models can use it as long as it's able to join to the model they're working with. Therefore, they can be used in multiple models because they are saved to your One Model, not to the model they were originally built in. This is super awesome because it makes it easy to share work and to build an impressive library of generative attributes quickly. You can search for generative attributes by typing free text into the search bar like so. You can also edit existing generative attributes by clicking the pencil icon, or you can delete generative attributes by clicking the trash icon.

Please be aware that if you edit a generative attribute that a model on your site is using, you will impact that model. If that is the case, you will see a warning in the edit screen stating "This generative attribute is being used in other augmentations. Changing it here will change the results of these augmentations." Similarly, if you try to delete a generative attribute that is being used in other models, you will get this warning that tells you which models it's being used in. You can still delete it if you choose, but doing so will impact those models.

Chapter 5

Components to Build a Generative Attribute

Now we will go through the components to build a generative attribute. To create a new generative attribute, click the 'Create' button here on the right side of the screen.

The first component is the existing One Model metric that you want to use. All base metrics that are able to join to this model will be available in this dropdown. Calculated metrics are not available to use when building generative attributes. You can always create more metrics in Explore if you wish, and then come back and create new generative attributes with them.

The next component is the time selection. The time selection is looking backwards from the sample date in the base query, which is the population date in step 2 of the recipe. There are three choices available in the dropdown.

First, we have 'Same as Base Query', which when selected means the time frame selection from the base query of the model is used. This uses the selections for the time interval from the "How far into the future do you want to predict?" step in the recipe, which is defaulted to one year.

Next, we have 'All time', which is exactly what it sounds like. And when selected means the generative attribute will include all time going backward from the sample date.

And finally, we have 'Explicit', which allows an explicit time frame that can differ from the base query to be selected.

First, you will be prompted to select year, quarter, or month in this dropdown, and then you can input how many you would like with free text in the next field. You can also use these arrows to increase or decrease the number of time intervals. For example, if we wanted it to be 6 quarters, we would select quarter here and then type a '6' like so. And that would mean that the generated attribute would span for 6 quarters.

The next component is our 'Group By' value. Group by is the aggregation level of your data. By default, generative attributes will be the same granularity as the dataset key of your base query. This is most commonly at the individual employee level with a unique identifier of person ID from the employee table. However, you can aggregate this to any column available in the fact table from which the population metric from the base query is created. We have two choices here. First, we have 'Same as Base Query', which as I just described is grouping the generative attribute by the unique identifier. For example, a promotion's attribute would be how many promotions that individual employee has had in the time selection. And next, we have 'Explicit', which allows you to aggregate with any column from the same table as your unique identifier. You will be prompted to pick a 'Group By' column by either searching here or scrolling.

Common aggregation columns include team, which might be grouped by manager ID or supervisor ID, like so, or location or plant. For example, a promotions attribute could be a team promotions, which should be how many promotions were on the entire team, and that could be grouped by that manager ID that we had just pulled up.

Next, we have the option to filter the generative attribute by adding a filter dimension. You are filtering the metric you selected in the metric component step. You can search for a specific dimension by typing here or scroll to find the dimension you wish to filter with. The filter function works the same here as it does in Explore and Storyboards. For example, you could filter a promotions generative attribute by event reason to only include promotions that are due to the progression reason like so. If you wish to get rid of a filter on a generative attribute, click on the trash can icon.

Next, you have the option to fill nulls with zeros by checking this box. Checking this box means that if the value is null or an empty string, it will become a 0. This is helpful so that your generative attributes are not dropped from the model for being too null without the model actually getting a chance to try them.

We also have the option to convert to binary by checking this box. If this box is checked, nulls, empty strings, and 0s get grouped together as 0 and everything else becomes a 1. This is helpful if you just want to measure if something has occurred or not versus how many times are on what scale.

And finally, you will see the 'Generative Attribute Name'. These are generated by One AI and can get a little messy depending on how your generative attribute was built and how much customization you did. I recommend checking the 'Use Custom Name' box so that you can give your generative attributes a name that makes sense and is easy to understand without users having to look into how it was built.

For example, we could call this one "Team Promotions (6 Quarters) - Binary", so that users know that we're using a promotions metric, it's been grouped at a team level, it spans 6 quarters, and we have converted it to binary. Then when you're happy with your generative attribute, you can click 'Save'. This will bring you back to the recipe screen where you can click the selection box, which is here. And by doing so, that will bring it into your model that you are working with if you wish to do so.

Chapter 6

Step-by-Step Generative Attribute Building Examples

Section 5: Step-by-Step Generative Attribute Building Examples

If you are not interested in or ready to see examples of how to build the different types of generative attributes, you can skip to Section 6 - Validating Generative Attributes in One AI.

Now that we've gone through the components needed to build a generative attribute, let's actually build a few together. Click the 'Create' button to get started.

First, we will build a metric generative attribute that counts how many promotions each employee has received.

For our metric, we're going to use our promotions metric.

For our time selection, we will leave it as 'Same as Base Query', which means it will count promotions from the last year because we are using the default settings for this recipe. We are also going to leave our 'Group By' as 'Same as Base Query' because our unique id for this recipe is employee ID and we want the aggregation to be at the employee level.

We are not going to filter this generative attribute, so we will also skip the filter step.

We will check the 'Fill NULLS with 0' box because if employees have not received a promotion, they will have a null value here, and we don't want the generative attribute to be automatically dropped for being too null.

And then because this is just called promotions, users won't know or have much information on how this generative attribute was built. So we're going to use a custom name and we will call it "Employee Promotions (Same as base) - Null-filled" to give users a little bit better idea of how it was built without having to actually click into it. And once we're happy with it, we'll click 'Save'.

Now we will build a customized metric generative attribute that measures how many promotions an employee has received due to succession for all time converted to binary.

So for our metric, again, we're gonna grab that promotions metric that we used in the last example, and we're gonna update our time selection from same as base query to all time. We're going to leave our group by at same as base query because, again, we want this to be on the employee level.

For filters. We're going to grab an event reason filter so that we can filter our promotions reason to only include succession promotions.

And then we're not going to fill nulls with zeros because we're going to convert to binary. We're doing this because I want to measure if employees have ever received this type of promotion, and I also want to avoid the attribute being automatically dropped for being too null. So converting to binary will accomplish both of those.

And then again, we're going to use a custom name to give users a good idea of how this generative attribute was built. So we're going to call this one "Employee Succession Promotions (All time) - Binary".

And then once you've had a chance to look it over and are happy with it, you can click 'Save'.

For our last example, we will build an aggregation generative attribute that measures team headcount or team size. So for the metric, we're going to grab a headcount EOP metric.

And for our time selection, we're going to leave it the 'Same as base', but our 'Group by', we will be updating to explicit. And for our group by column, we will use manager ID because that's how teams are defined in our demo site. We're not going to filter or fill nulls with zeros or convert to binary.

For our generative attribute name, of course, we're going to give it a custom name so that users know how it was built. And we're going to call this one "Team Headcount (Same as base)".

Again, take a look to make sure it's built correctly before clicking 'Save'.

Chapter 7

Validating Generative Attributes in One AI

Section 6 - Validating Generative Attributes in One AI

Once you've created all of the generative attributes that you wish to, you will use the selection boxes to select which ones you want the model to try. You can select them individually by clicking their accompanying selection box, like so, or you can click the selection box at the top to select all generative attributes.

Before we run the model again with these new attributes, we should validate our selections to ensure data integrity. This can be done by expanding the "Would you like to verify all of the selections you have made are valid?" section of the recipe, and then by clicking the 'Generate Data Statistics' button.

A report containing statistics about all of the data the recipe generates will be displayed. Please note that this may take a few minutes to load. Once validation is complete, you will get a message at the top that either says "Action is needed" in red that will list the errors, or it will say "Success" in green, meaning there are no issues with your data model that would cause the query to fail, and you can proceed to run your model when you wish to.

You can see info about each attribute included in the model, such as non-null count, mins, maxes, averages, standard deviations, etcetera. Generative attributes should be listed near the top of this report. Once you've finished validating your model, you can save by clicking the save icon in the upper right corner, and then save again from the augmentation screen by scrolling to the bottom.

Then you can rerun your model to let these powerful features go to work. Once your model is complete, check out your EDA report to see if they were selected in your model and their impact.

Chapter 8

Conclusion & Thanks

In summary, generative attributes are an important tool in One AI machine learning, offering new ways to enhance model performance and provide the model with additional more meaningful information that might not be explicitly captured by the original features.

You have learned about the three key types of generative attributes available in One AI, metric based attributes, customized metric based attributes, and aggregated attributes. You should have practical insights into creating and managing these attributes within the One AI platform, emphasizing the importance of validation to ensure data integrity and model readiness.

By leveraging generative attributes effectively, you can enrich their feature sets, improve model accuracy, and gain deeper insights from the data. Check out our other modules in the recipes section to continue learning how to optimize your machine learning models and uncover interesting insights. Happy modeling!