“Feature Engineering” Module Transcript

Chapter 1

Intro, Topics Covered, & Learning Outcomes

Hi all. My name is Hayley, and I'm on the One AI team here at One Model. In this module, we are going to build upon what you learned in the "AutoML" and "Data Pre-processing" module by diving into feature engineering.

We will cover what a feature is and the different types of features to lay the groundwork to discuss feature engineering. Then, we will move into what feature engineering is and its significance in machine learning, the process and techniques of feature engineering, and using exploratory data analysis for feature exploration and engineering.

After completing this module, you will have the ability to clearly explain what a feature is, particularly in the context of machine learning, and distinguish between the different types. You will understand the importance of feature engineering and its techniques such as data cleaning, feature selection, and feature transformation. And, you will be confident in conducting exploratory data analysis to understand dataset characteristics & inform feature engineering processes.

Chapter 2

Feature Overview

Section 2 - Feature Overview

Before exploring feature engineering, it's important to understand what a feature is in the context of machine learning.

A feature is an individual, measurable property or characteristic of the data used as input for a machine learning model to make predictions and classifications.

Features are also known as input variables, predictors, columns, or attributes.

You will be exposed to many of these somewhat interchangeable terms throughout One AI Academy and in relevant documentation in the help center.

Features can take various forms depending on the data type and the problem being solved. The main types of features are as follows.

Numerical features are continuous variables that represent quantities or measurements such as age, salary, or temperature.

Date features are a subset of numerical features such as date of hire or date of birth.

Categorical features are non-numerical information divided into discrete categories or labels such as city or gender.

Categorical features can be ordinal, which means they have a clear ordering or ranking such as low, medium, or high, or nominal, which means there is no inherent order such as colors or types of animals.

Binary features are a specific type of categorical feature with only two possible values, such as 0 or 1 or yes or no.

And finally, generated or derived features are created from existing data through transformations, combinations, aggregations, or adjusting time periods to be different from the model.

Examples include "Team Headcount" or "Count of Promotions for All Time". In One AI, we call these generative attributes. We will discuss them in more detail later in this module and in the generative attribute module.

Chapter 3

Feature Engineering Overview

Section 3 - Feature Engineering Overview

Raw data is rarely in a format that a machine learning model can use directly. Machine learning algorithms require data to be in a structured format they can understand and process often as numerical values or vectors instead of text or categorical values.

Therefore, it needs to be extracted and transformed into features through a process called feature engineering.

You can get a good idea of where this specific process fits in the overall machine learning process in this chart.

Effective feature engineering significantly improves model performance and generalization on new data by allowing the model to focus on important relevant information while ignoring noise.

This also enhances the model's computational efficiency without sacrificing performance by strategically reducing dimensionality with proven feature engineering techniques instead of at random.

Well engineered features result in more interpretable models. When features are carefully chosen or transformed to represent meaningful data aspects, it becomes easier to understand how the model's predictions are influenced, leading to more meaningful and actionable insights.

Chapter 4

Feature Engineering Process & Techniques

Section 4 - Feature Engineering Process & Techniques

Feature Engineering involves a series of steps using multiple techniques to create, select, or transform input features to build strong machine learning models. Here are the typical steps and techniques.

Step 1 - understanding the data

It's critical that the model creator has a full understanding of the model dataset and the problem domain, which is what you are trying to accomplish with the model.

This involves communicating regularly with stakeholders and data exploration techniques such as identifying the types of features available, understanding their distributions, and gaining insights into potential relationships between features and the target variable, which is what you are predicting on. You can learn more about this process in the data exploration module.

Step 2 - data cleaning

Data cleaning involves handling missing and constant values, checking for correlations, and detecting data leakage.

This may involve techniques such as null filling - which is also known as imputation - for missing values, removing outliers, and correcting data errors. You can learn more about this in the data preprocessing module.

Step 3 - feature selection

Feature selection involves selecting the most relevant features while discarding irrelevant ones or redundant ones. This process helps reduce dimensionality and prevent overfitting.

While a robust dataset with a variety of features is important, it's crucial to find the sweet spot between having enough features for solid training and accurate predictions, while avoiding excessive complexity that makes models hard to interpret.

Some key techniques include univariate feature selection, which involves selecting features using statistical tests like ANOVA or chi square to assess each feature's relationship with the target variable independently.

One AI employs univariate testing during the first phase of dimensionality reduction to rank features based on their individual relevance.

There's also recursive feature elimination, which is an iterative algorithm that systematically removes the least important features until the optimal number of features is reached, as determined by the best model performance.

One AI uses RFE during the second phase of dimensionality reduction.

And then there are multiple techniques that can be used for determining feature importance.

Common methods include permutation importance or feature importance plots. Permutation importance checks how important a feature is by calculating how much the model's error changes when we randomly shuffle that feature's values.

Feature importance plots visually display the importance of each feature, making it easy to see which features contribute most to the model. In One AI, the method we use depends on the type of model we're working with.

For example, for tree based algorithms, we measure feature importance by looking at how much a feature helps reduce a certain criterion, like entropy.

If we're using entropy as our criterion, the importance of a feature is based on how much it reduces entropy. This reduction is calculated across the entire decision tree or averaged across all trees if we're using a random forest.

Step 4 - feature transformation

Feature transformation involves modifying features to make them more suitable for consumption by the machine learning algorithms.

Key techniques include scaling, which is the mathematical transformation of all the numerical features in a dataset to a common scale to ensure they are on the same scale and appropriately weighted by the algorithm. This avoids bias in your models.

Scaling is also known as normalization.

Next, we have one hot encoding (OHE), which involves converting categorical variables into binary columns to eliminate bias and make them consumable by the algorithm.

There is also log transformation, which is used to address data skewedness, and power transformation, which stabilizes variance.

These techniques and much more are discussed in detail in the preprocessing module.

Step 5 - feature creation

The feature creation step involves generating new features from existing ones to capture additional information or relationships in the data that otherwise may have been missed. This can be done through various techniques. To start, we have generative attributes, which are new input variables derived from your original dataset, enabling data aggregation in various ways. For example, you can compare team or department metrics to individual employee attributes.

Additionally, you can adjust the time span of metrics such as creating all time or explicit time metrics. For more details, refer to our Generative Attribute module.

Next we have binning, which involves grouping continuous features into discrete categories. For example, Salary ranges.

Next, we have interaction terms, which can be created by capturing the interactions between existing features. And finally, we have polynomial features, which can be created by combining existing features through multiplication.

Step 6 - iterate and improve

Effective feature engineering requires a deep understanding of the data and problem domain along with experimentation to identify the most effective features.

This iterative process involves refining features based on model performance feedback using cross validation, which is covered in an upcoming module, and fine tuning your feature pipeline. Regular monitoring and adjustments are essential as data evolves over time.

Chapter 5

Exploratory Data Analysis & Feature Engineering

Section 5 - Exploratory Data Analysis and Feature Engineering

Exploratory data analysis is a foundational step in feature engineering providing the necessary insights and understanding to effectively transform and engineer features for machine learning models.

EDA is used to analyze model datasets and summarize their main characteristics, often employing data visualization methods.

When a One AI model runs successfully, an EDA report is generated to provide a deeper understanding of the dataset.

The report details which features were selected and which were automatically dropped with the automatic drop reasons, allowing you to adjust the model settings as needed.

You can also perform in-depth variable analysis with visualizations, see how each feature was transformed, visualize missing data, and download comprehensive correlation data to understand relationships among features.

It's an exceptional tool to use throughout the steps of feature engineering, helping you understand the data and what One AI is doing with it.

Chapter 6

Data Preprocessing for Individual Variables

Section 4 - Data Preprocessing for Individual Variables

One AI allows for configuration of preprocessing for individual variables with per column interventions.

Different columns within the same dataset may have distinct characteristics, such as varying data types, scales, distributions, and degrees of missing data or outliers.

Applying the same preprocessing to all features may not be optimal. This approach recognizes these differences between columns and optimizes the model's ability to learn and generalize patterns from the data, resulting in a more performant and understandable model.

Various treatments can be applied in One AI, including the following.

First, modifying the column or column's droppability with the choices of droppable, where One AI determines if it will be dropped based on predictive power, not droppable, where violating global settings will not result in the automatic dropping of a column, and always droppable, where the column will always be dropped.

Next, configuring how you want One AI to handle nulls for the selected column or columns. We have several null fill strategies to include mean, median, mode, backwards fill, forwards fill, pad, or you can choose a custom value to fill all nulls with.

And finally, performing type specific interventions to include categorical interventions that allow users to specify nodes to exclude from being considered as features and or specify nodes to force select as features. Additionally, continuous interventions allow users to force select variables as features and change the continuous strategy for a specific column to bin or scale. Reminder that the default is scale unless changed in the global settings.

If changed to bin here, you can specify the bin type as well. We go into more detail about these interventions to include the how to in the per column interventions module.

Chapter 7

Conclusion & Thanks

In conclusion, feature engineering is an essential component of machine learning. In this module, we have covered the core concepts and techniques of feature engineering from data understanding to refinement.

Remember, well engineered features enhance model performance and interpretability, providing meaningful insights. As you embark on your feature engineering journey, embrace experimentation and iteration to refine and adapt your models to your organization's evolving needs. Happy modeling!