**Tyler Boda**

In the module on data preprocessing and feature engineering, the code implements crucial steps that turn raw, inconsistent data into a usable format for machine learning models. Initially, it recognizes missing values and applies appropriate methods to fill them based on the data type like replacing missing numerical values with averages or filling categorical data with the most frequent item.

Next, the code converts categorical variables into numbers using two techniques: ordinal encoding, which assigns numbers to categories with a natural order, and one hot encoding, which creates flags for categories without order. This ensures that machine learning algorithms (which only process numeric inputs) can correctly interpret the data without misunderstanding the relationships.

Other than the basic techniques, the module also showed the importance of understanding the context of the data when choosing preprocessing strategies. Choosing the right imputation method involves considering the knowledge of the missing data and its potential impact on the overall analysis. The encoding methods impact how the machine learning model interprets the data. Incorrect encoding can lead to misleading results or poor model performance. The module emphasized experimenting with different preprocessing techniques and visually checking the data after each step. It also introduced the concept of pipeline, which automates the entire preprocessing sequence, reducing human error and improving repeatability.

The module then scales and normalizes numerical features to make sure that variables measured in different units or scales do not wrongfully dominate the model’s learning process. Outlier detection methods identify unusually high or low values that could outweigh modeling. Instead of dropping these outliers, the code ‘caps’ them to keep the dataset consistent. Feature engineering creates new variables and connects them to existing ones to show hidden patterns and improve the model’s ability to predict outcomes.

All these steps are organized into a process. A reusable workflow that makes sure that it becomes consistent, efficient, and reproducible data preparation whether used on training data or future unseen data. Throughout, visualizations like histograms, boxplots, and heatmaps help look for changes and confirm that each preprocessing step improves data quality.